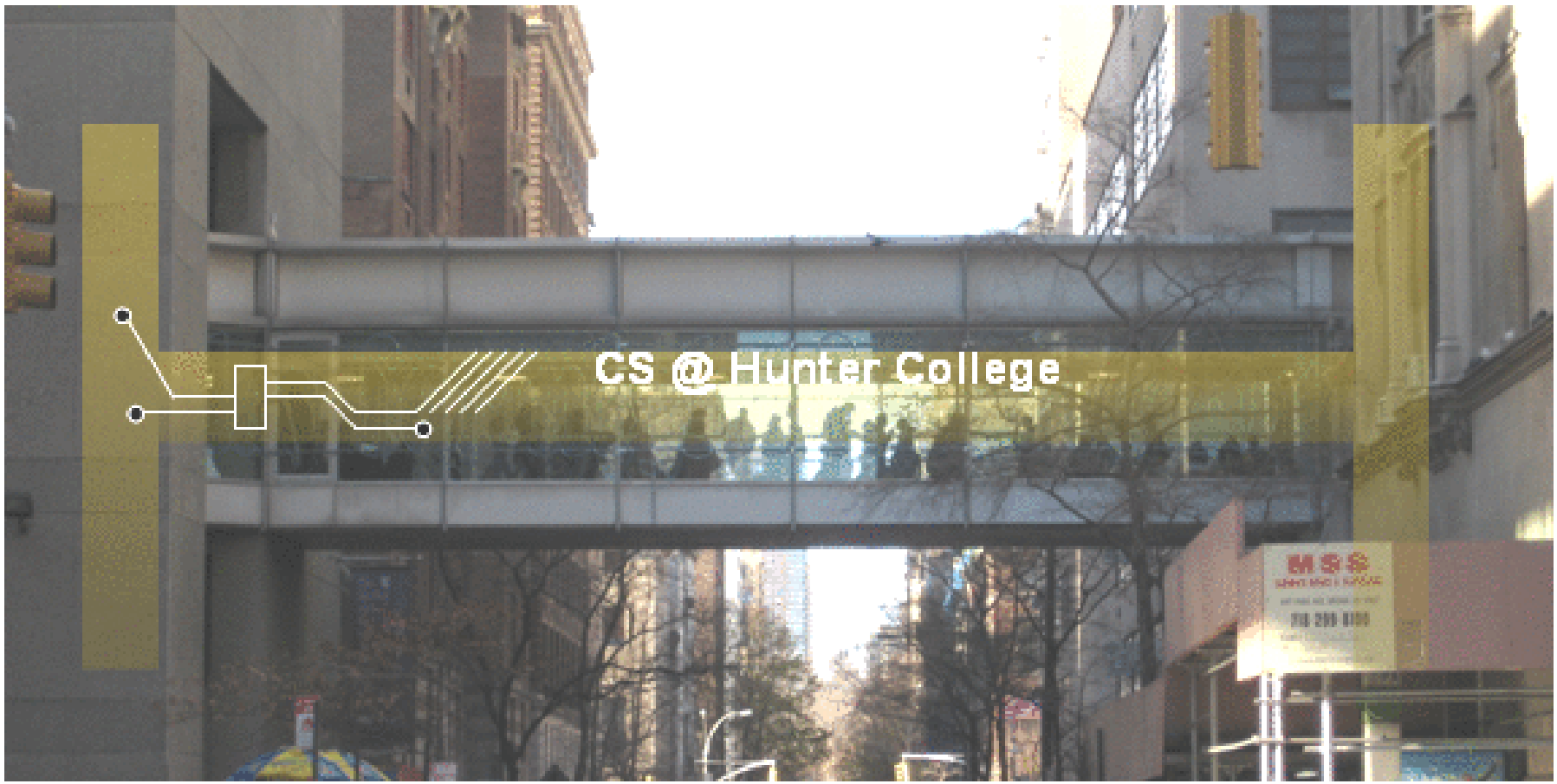
CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Announcements

M	T	W	Th	F
8/27 L1	8/28 L1	8/29 L1 Lecture 1	8/30 L1	8/31 L1
Labor Day	9/4 L2	M L2	9/6 L2	9/8 L2
No class	No class	9/12 L2 Lecture 2	9/13 L3	9/14 L3
9/17 L3	No class	No class	9/20 L4	9/21 L4
9/24 L4	9/25 L3	9/26 L3 Lecture 3	9/27 L5	9/28 L5
10/1 L5	10/2 L4	10/3 L4 Lecture 5	10/4 L6	10/5 L6
Columbus Day	10/9 L5	10/10 L5 Lecture 5	10/11 L7	10/12 L7
10/15 L6	10/16 L6	10/17 L6 Lecture 6	10/18 L8	10/19 L8
10/22 L7	10/23 L7	10/24 L7 Lecture 7	10/25 L9	10/27 L9
10/29 L8	10/30 L8	10/31 L8 Lecture 8	11/1 L10	11/2 L10
11/5 L9	11/6 L9	11/7 L9 Lecture 9	11/8 L11	11/9 L11
11/12 L10	11/13 L10	11/14 L10 Lecture 10	11/15 L12	11/16 L12
11/19 L11	11/20 L11	11/21 L11 Lecture 11	TG Recess	TG Recess
11/26 L12	11/27 L12	11/28 L12 Lecture 12	11/29 L13	11/30 L13
12/3 L13	12/4 L13	12/5 L13 Lecture 13	12/6 L14	12/7 L14
12/10 L14	12/11 L14	12/12 L14 Lecture 14	Reading Day	

- No more holidays until October!

Announcements

M	T	W	Th	F
8/27 L1	8/28 L1	8/29 L1 Lecture 1	8/30 L1	8/31 L1
Labor Day	9/4 L2	M L2	9/6 L2	9/8 L2
No class	No class	9/12 L2 Lecture 2	9/13 L3	9/14 L3
9/17 L3	No class	No class	9/20 L4	9/21 L4
9/24 L4	9/25 L3	9/26 L3 Lecture 3	9/27 L5	9/28 L5
10/1 L5	10/2 L4	10/3 L4 Lecture 5	10/4 L6	10/5 L6
Columbus Day	10/9 L5	10/10 L5 Lecture 5	10/11 L7	10/12 L7
10/15 L6	10/16 L6	10/17 L6 Lecture 6	10/18 L8	10/19 L8
10/22 L7	10/23 L7	10/24 L7 Lecture 7	10/25 L9	10/27 L9
10/29 L8	10/30 L8	10/31 L8 Lecture 8	11/1 L10	11/2 L10
11/5 L9	11/6 L9	11/7 L9 Lecture 9	11/8 L11	11/9 L11
11/12 L10	11/13 L10	11/14 L10 Lecture 10	11/15 L12	11/16 L12
11/19 L11	11/20 L11	11/21 L11 Lecture 11	TG Recess	TG Recess
11/26 L12	11/27 L12	11/28 L12 Lecture 12	11/29 L13	11/30 L13
12/3 L13	12/4 L13	12/5 L13 Lecture 13	12/6 L14	12/7 L14
12/10 L14	12/11 L14	12/12 L14 Lecture 14	Reading Day	

- No more holidays until October!
- Google Engineering Info Sessions (10/3 or 10/10). More info at: `googleinfosession.splashthat.com`

Announcements

M	T	W	Th	F
8/27 L1	8/28 L1	8/29 L1 Lecture 1	8/30 L1	8/31 L1
Labor Day	9/4 L2	M L2	9/6 L2	9/8 L2
No class	No class	9/12 L2 Lecture 2	9/13 L3	9/14 L3
9/17 L3	No class	No class	9/20 L4	9/21 L4
9/24 L4	9/25 L3	9/26 L3 Lecture 3	9/27 L5	9/28 L5
10/1 L5	10/2 L4	10/3 L4 Lecture 5	10/4 L6	10/5 L6
Columbus Day	10/9 L5	10/10 L5 Lecture 5	10/11 L7	10/12 L7
10/15 L6	10/16 L6	10/17 L6 Lecture 6	10/18 L8	10/19 L8
10/22 L7	10/23 L7	10/24 L7 Lecture 7	10/25 L9	10/27 L9
10/29 L8	10/30 L8	10/31 L8 Lecture 8	11/1 L10	11/2 L10
11/5 L9	11/6 L9	11/7 L9 Lecture 9	11/8 L11	11/9 L11
11/12 L10	11/13 L10	11/14 L10 Lecture 10	11/15 L12	11/16 L12
11/19 L11	11/20 L11	11/21 L11 Lecture 11	TG Recess	TG Recess
11/26 L12	11/27 L12	11/28 L12 Lecture 12	11/29 L13	11/30 L13
12/3 L13	12/4 L13	12/5 L13 Lecture 13	12/6 L14	12/7 L14
12/10 L14	12/11 L14	12/12 L14 Lecture 14	Reading Day	

- No more holidays until October!
- Google Engineering Info Sessions (10/3 or 10/10). More info at:
  `googleinfosession.splashthat.com`
- Starting this week, each lecture will include a survey of computing research and the tech industry in NYC.

Announcements

M	T	W	Th	F
8/27 L1	8/28 L1	8/29 L1 Lecture 1	8/30 L1	8/31 L1
Labor Day	9/4 L2	M L2	9/6 L2	9/8 L2
No class	No class	9/12 L2 Lecture 2	9/13 L3	9/14 L3
9/17 L3	No class	No class	9/20 L4	9/21 L4
9/24 L4	9/25 L3	9/26 L3 Lecture 3	9/27 L5	9/28 L5
10/1 L5	10/2 L4	10/3 L4 Lecture 5	10/4 L6	10/5 L6
Columbus Day	10/9 L5	10/10 L5 Lecture 5	10/11 L7	10/12 L7
10/15 L6	10/16 L6	10/17 L6 Lecture 6	10/18 L8	10/19 L8
10/22 L7	10/23 L7	10/24 L7 Lecture 7	10/25 L9	10/27 L9
10/29 L8	10/30 L8	10/31 L8 Lecture 8	11/1 L10	11/2 L10
11/5 L9	11/6 L9	11/7 L9 Lecture 9	11/8 L11	11/9 L11
11/12 L10	11/13 L10	11/14 L10 Lecture 10	11/15 L12	11/16 L12
11/19 L11	11/20 L11	11/21 L11 Lecture 11	TG Recess	TG Recess
11/26 L12	11/27 L12	11/28 L12 Lecture 12	11/29 L13	11/30 L13
12/3 L13	12/4 L13	12/5 L13 Lecture 13	12/6 L14	12/7 L14
12/10 L14	12/11 L14	12/12 L14 Lecture 14	Reading Day	

- No more holidays until October!
- Google Engineering Info Sessions (10/3 or 10/10). More info at:
  `googleinfosession.splashthat.com`
- Starting this week, each lecture will include a survey of computing research and the tech industry in NYC.
- Today: Prof. William Sakas, Chair of Computer Science: Computational Linguistics.

Frequently Asked Questions

From lecture slips & recitation sections.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines. Let us know– some we can fix, some have to match exactly.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines. Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines. Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions:

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")`

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string:

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]`

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.
- Could you explain more about arithmetic (especially modulo!) in Python?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.
- Could you explain more about arithmetic (especially modulo!) in Python?
  Yes, will do!

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.
- Could you explain more about arithmetic (especially modulo!) in Python?
  Yes, will do!
- One more time on all the `range()` options?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.
- Could you explain more about arithmetic (especially modulo!) in Python?
  Yes, will do!
- One more time on all the `range()` options?
  We'll have some in group work and a quick review.
- Could we do more final problems?

Frequently Asked Questions

From lecture slips & recitation sections.

- I didn't turn in a lecture slip last week. Did I miss it?
  No worries. If you miss a lecture slip, your final exam grade replaces that score.
- Can I get a copy of the lecture slides and programs from lecture?
  Yes, the slides are posted on the class website.
- I'm sure I did Problem 9 correctly, but Gradescope disagrees. Why?
  Some of the grading scripts are really finicky about spacing and new lines.
  Let us know– some we can fix, some have to match exactly.
- Why do we do problems on things we haven't learned yet?
  You're more likely to pass a 1st year science course if you had it in high school.
  Next best thing: "pre-test" to give a "mental scaffold" to hold new ideas.
- What's the difference between the parenthesis and the brackets?
  Parenthesis are for functions: `print("Hi!")` *or* `tori.left(90)`
  Brackets are for accessing part of a list or string: `words[1]` *or* `mess[3:6]`.
- Could you explain more about arithmetic (especially modulo!) in Python?
  Yes, will do!
- One more time on all the `range()` options?
  We'll have some in group work and a quick review.
- Could we do more final problems?
  We'll start with some today since we didn't get to them last lecture.

Today's Topics

- Arithmetic
- Indexing and Slicing Lists
- Colors
- Hexadecimal Notation
- 2D Arrays & Image Files

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).

Practice Quiz & Final Questions



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).
- We're starting with Spring 2018, Mock Exam.

Arithmetic

Some arithmetic operators in Python:

- Addition:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:
  `weeks = totalDays // 7`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:
  `weeks = totalDays // 7`
- Remainder or Modulus:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:
  `weeks = totalDays // 7`
- Remainder or Modulus:
  `days = totalDays % 7`

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:
  `weeks = totalDays // 7`
- Remainder or Modulus:
  `days = totalDays % 7`
- Exponentiaion:

Arithmetic

Some arithmetic operators in Python:

- Addition: `sum = sum + 3`
- Subtraction: `deb = deb - item`
- Multiplication: `area = h * w`
- Division: `ave = total / n`
- Floor or Integer Division:
  `weeks = totalDays // 7`
- Remainder or Modulus:
  `days = totalDays % 7`
- Exponentiaion:
  `pop = 2**time`

In Pairs or Triples...

Mostly review:

```
 1  for d in range(10, 0, -1):
 2      print(d)
 3  print("Blast off!")
 4
 5  for num in range(5,8):
 6      print(num, 2*num)
 7
 8  s = "City University of New York"
 9  print(s[3], s[0:3], s[:3])
10  print(s[5:8], s[-1])
11
12  names = ["Eleanor", "Anna", "Alice", "Edith"]
13  for n in names:
14      print(n)
```

Python Tutor

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

(Demo with pythonTutor)

Review: range()

The three versions:

Review: `range()`

The three versions:

- `range(stop)`

Review: range()

The three versions:

- `range(stop)`
- `range(start, stop)`

Review: range()

The three versions:

- `range(stop)`
- `range(start, stop)`
- `range(start, stop, step)`

range()

What if you wanted to count by twos, or some other number:

```
1  #Predict what will be printed:
2
3  for num in [2,4,6,8,10]:
4      print(num)
5
6  sum = 0
7  for x in range(0,12,2):
8      print(x)
9      sum = sum + x
10
11 print(x)
12
13 for c in "ABCD":
14     print(c)
```

range()

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`

```
1  #Predict what will be printed:
2
3  for num in [2,4,6,8,10]:
4      print(num)
5
6  sum = 0
7  for x in range(0,12,2):
8      print(x)
9      sum = sum + x
10
11 print(x)
12
13 for c in "ABCD":
14     print(c)
```

range()

```
1  #Predict what will be printed:
2
3  for num in [2,4,6,8,10]:
4      print(num)
5
6  sum = 0
7  for x in range(0,12,2):
8      print(x)
9      sum = sum + x
10
11 print(x)
12
13 for c in "ABCD":
14     print(c)
```

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list: [start,start+step,start+2*step...,last] (where last is the largest start+k*step less than stop)

range()

```
1  #Predict what will be printed:
2
3  for num in [2,4,6,8,10]:
4      print(num)
5
6  sum = 0
7  for x in range(0,12,2):
8      print(x)
9      sum = sum + x
10
11 print(x)
12
13 for c in "ABCD":
14     print(c)
```

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list: [start,start+step,start+2*step...,last] (where last is the largest start+k*step less than stop)
- For example, if you want the the list [5,10,...,50] you would write:

range()

```
1  #Predict what will be printed:
2
3  for num in [2,4,6,8,10]:
4      print(num)
5
6  sum = 0
7  for x in range(0,12,2):
8      print(x)
9      sum = sum + x
10
11 print(x)
12
13 for c in "ABCD":
14     print(c)
```

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list:
  [start,start+step,start+2*step...,last]
  (where last is the largest start+k*step less than stop)
- For example, if you want the the list [5,10,...,50]
  you would write:

  `range(5,51,5)`

Slices

- Similar to `range()`, you can take portions or **slices** of lists and strings:

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

Slices

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

- Similar to `range()`, you can take portions or **slices** of lists and strings:

  `s[5:8]`

  gives: `"Uni"`

Slices

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

- Similar to `range()`, you can take portions or **slices** of lists and strings:

  `s[5:8]`

  gives: `"Uni"`
- Also works for lists:

Slices

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

- Similar to `range()`, you can take portions or **slices** of lists and strings:

  `s[5:8]`

  gives: `"Uni"`
- Also works for lists:

  `names[1:3]`

Slices

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

- Similar to `range()`, you can take portions or **slices** of lists and strings:

  `s[5:8]`

  gives: `"Uni"`
- Also works for lists:

  `names[1:3]`

  gives: `["Anna", "Alice"]`

Slices

```
1  for d in range(10, 0, -1):
2      print(d)
3  print("Blast off!")
4
5  for num in range(5,8):
6      print(num, 2*num)
7
8  s = "City University of New York"
9  print(s[3], s[0:3], s[:3])
10 print(s[5:8], s[-1])
11
12 names = ["Eleanor", "Anna", "Alice", "Edith"]
13 for n in names:
14     print(n)
```

- Similar to `range()`, you can take portions or **slices** of lists and strings:

  `s[5:8]`

  gives: `"Uni"`
- Also works for lists:

  `names[1:3]`

  gives: `["Anna", "Alice"]`
- Python also lets you "count backwards": last element has index: `-1`.

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:
    - ★ Black: 0% red, 0% green, 0% blue

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:
    - ★ Black: 0% red, 0% green, 0% blue
    - ★ White: 100% red, 100% green, 100% blue

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.
  - 8-bit colors: numbers from 0 to 255:

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.
  - 8-bit colors: numbers from 0 to 255:
    e.g. `(0, 255, 0)` is no red, 100% green, and no blue.

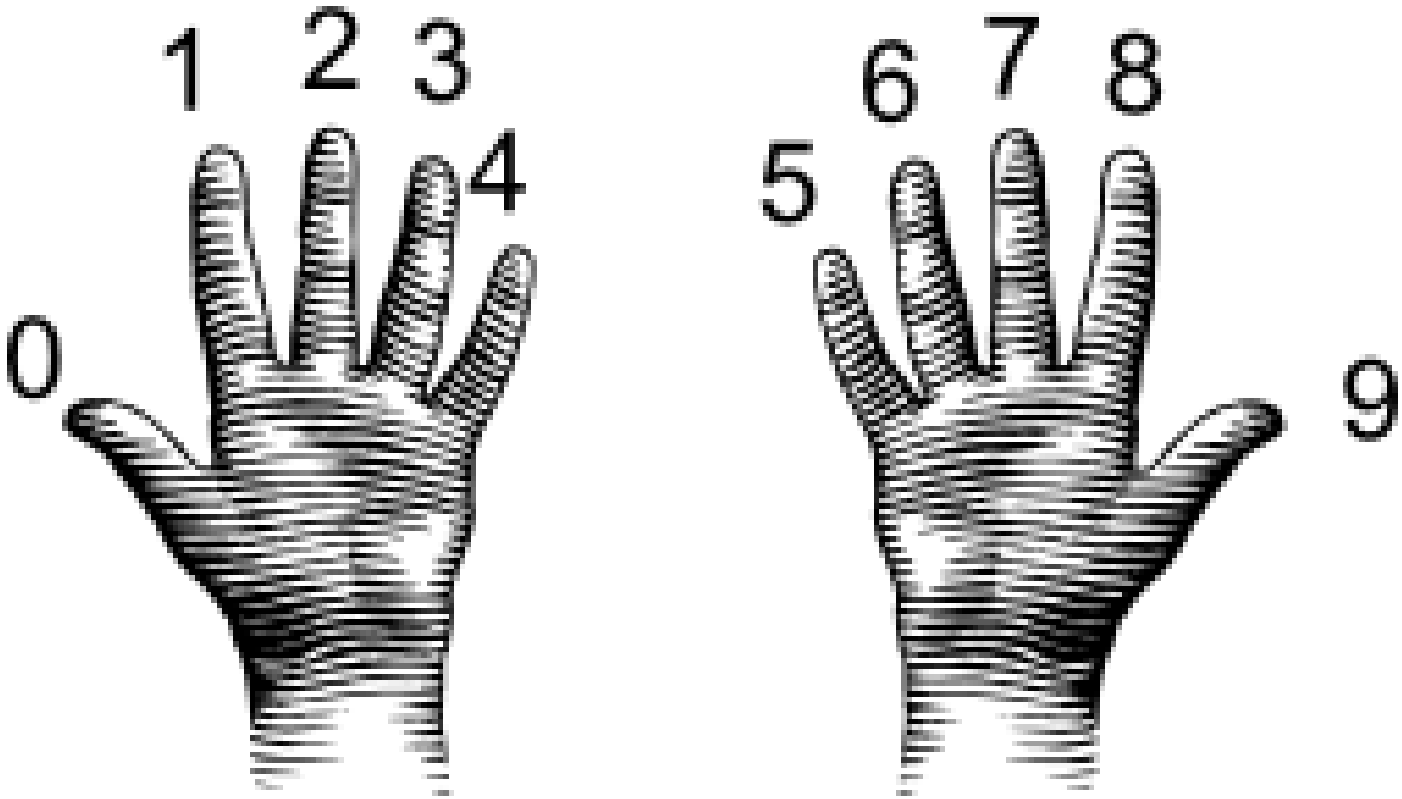
Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.
  - 8-bit colors: numbers from 0 to 255:
    e.g. `(0, 255, 0)` is no red, 100% green, and no blue.
  - Hexcodes (base-16 numbers)...

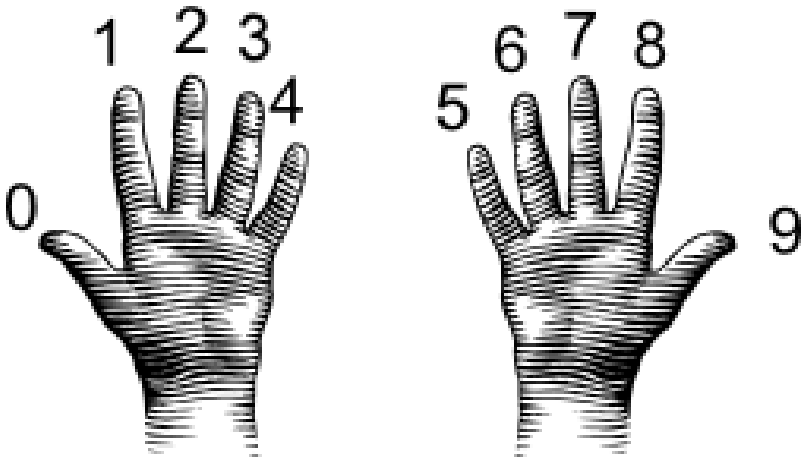
Decimal & Hexadecimal Numbers

Counting with 10 digits:



(from i-programmer.info)

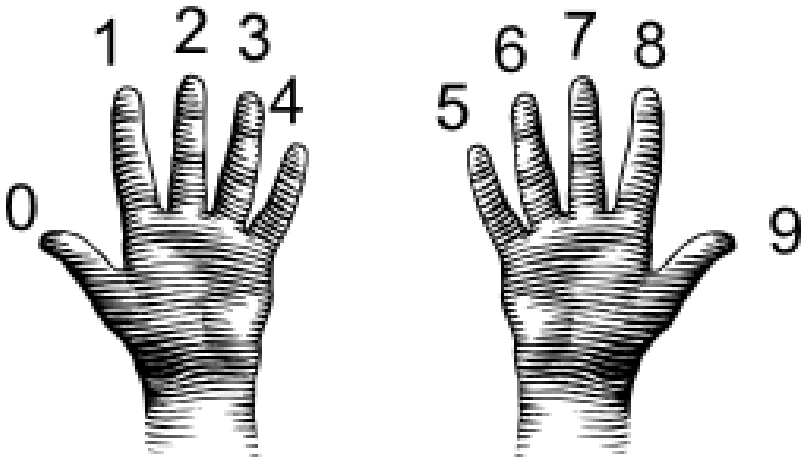
Decimal



(from i-programmer.info)

Decimal

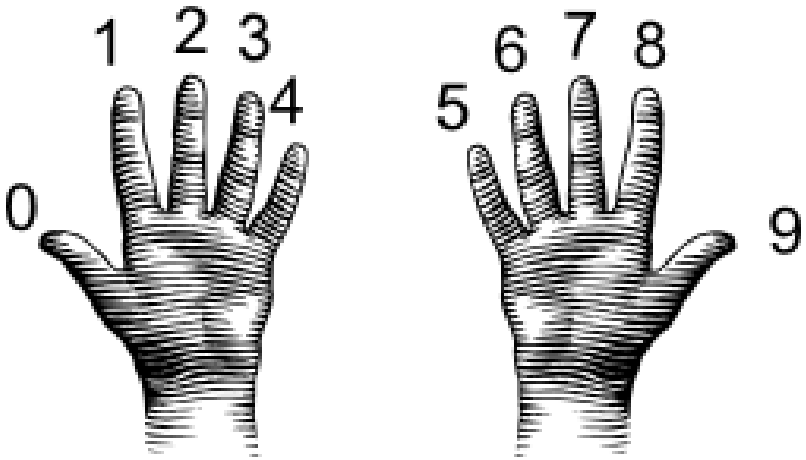
00 01 02 03 04 05 06 07 08 09



(from i-programmer.info)

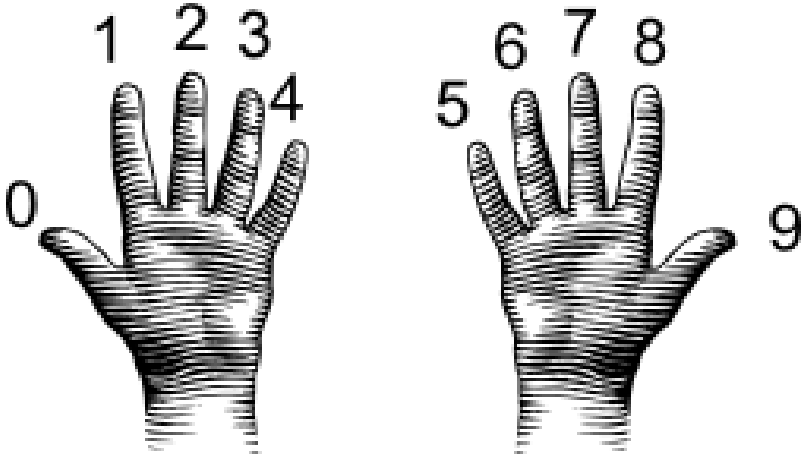
Decimal

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19



(from i-programmer.info)

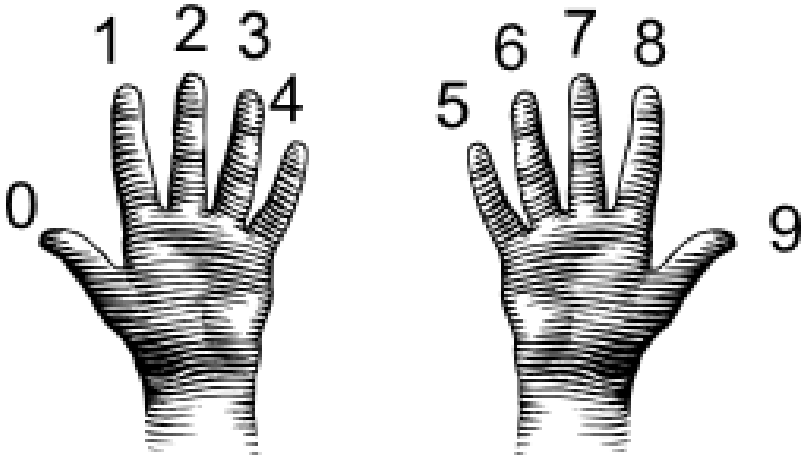
Decimal



(from i-programmer.info)

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29

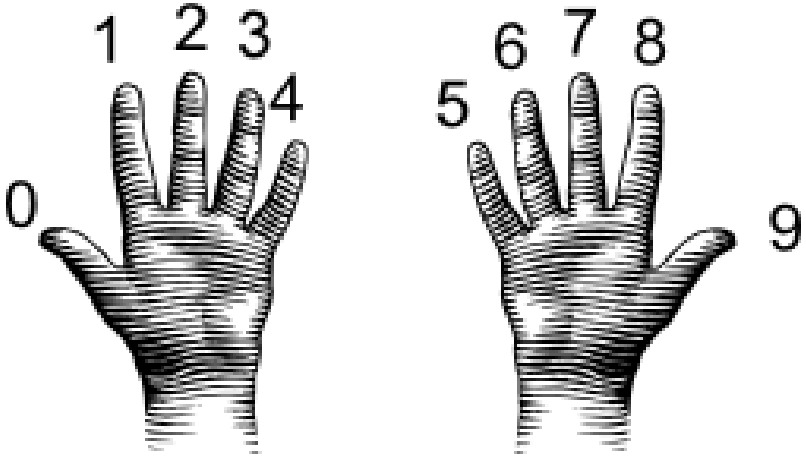
Decimal



(from i-programmer.info)

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39

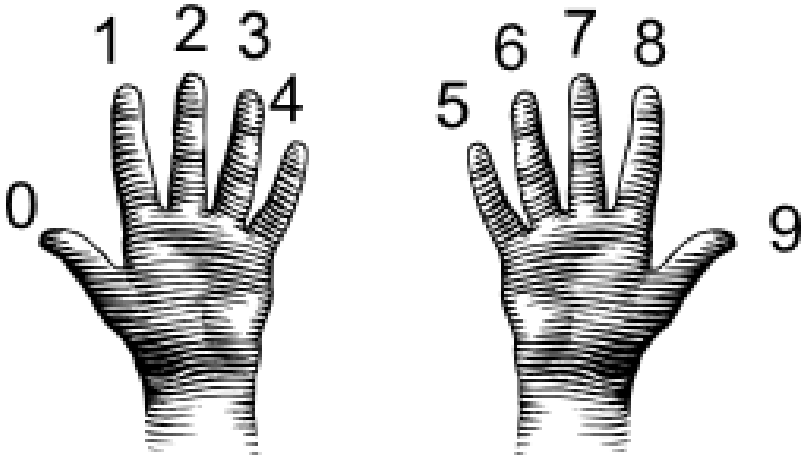
Decimal



(from i-programmer.info)

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49

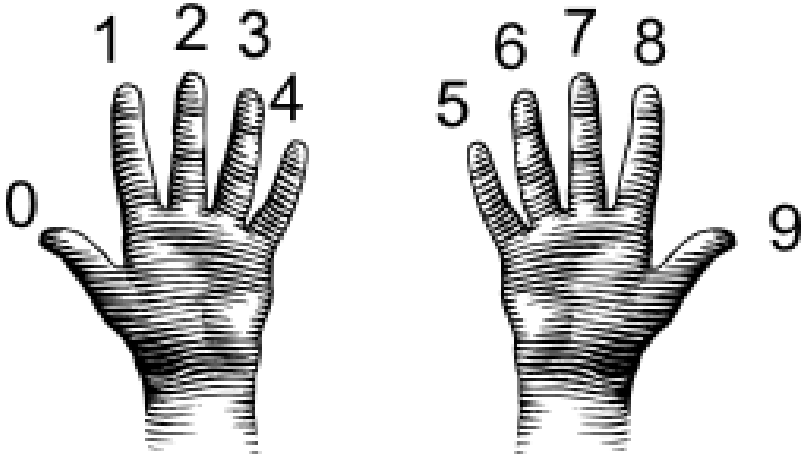
Decimal



(from i-programmer.info)

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59

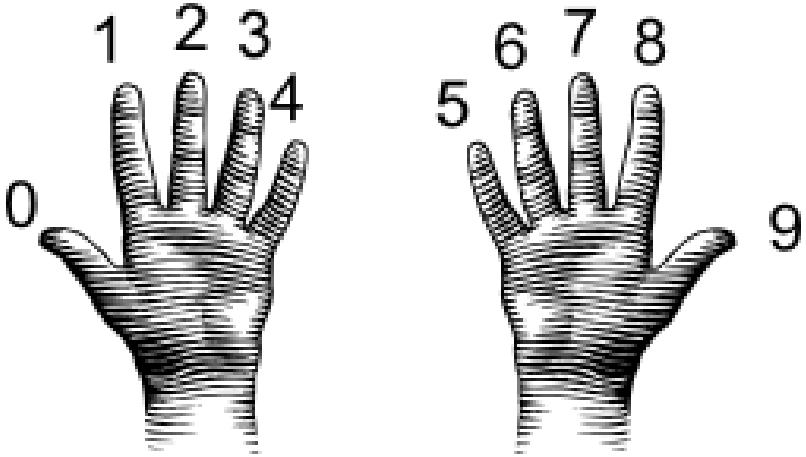
Decimal



(from i-programmer.info)

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69

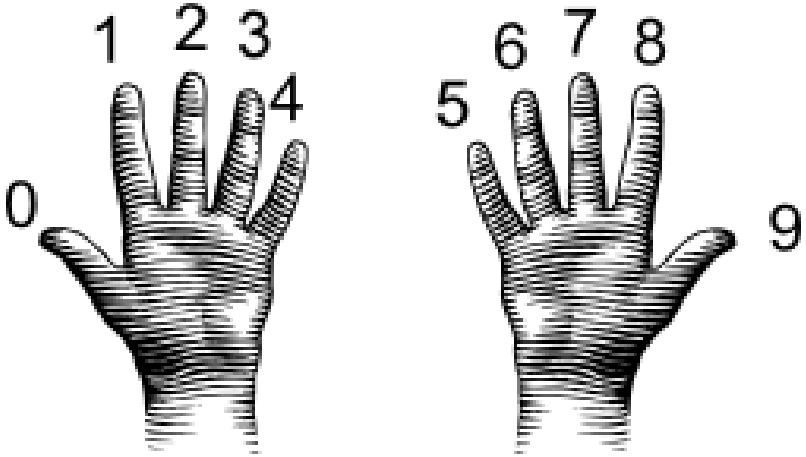
Decimal



(from i-programmer.info)

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79

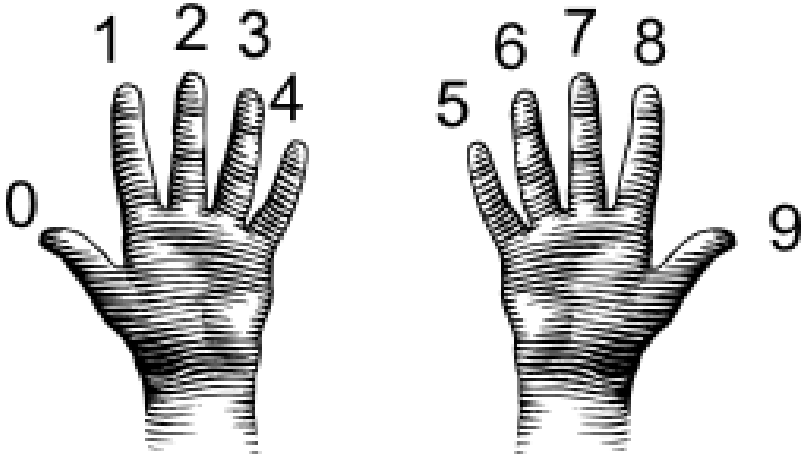
Decimal



(from i-programmer.info)

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89

Decimal

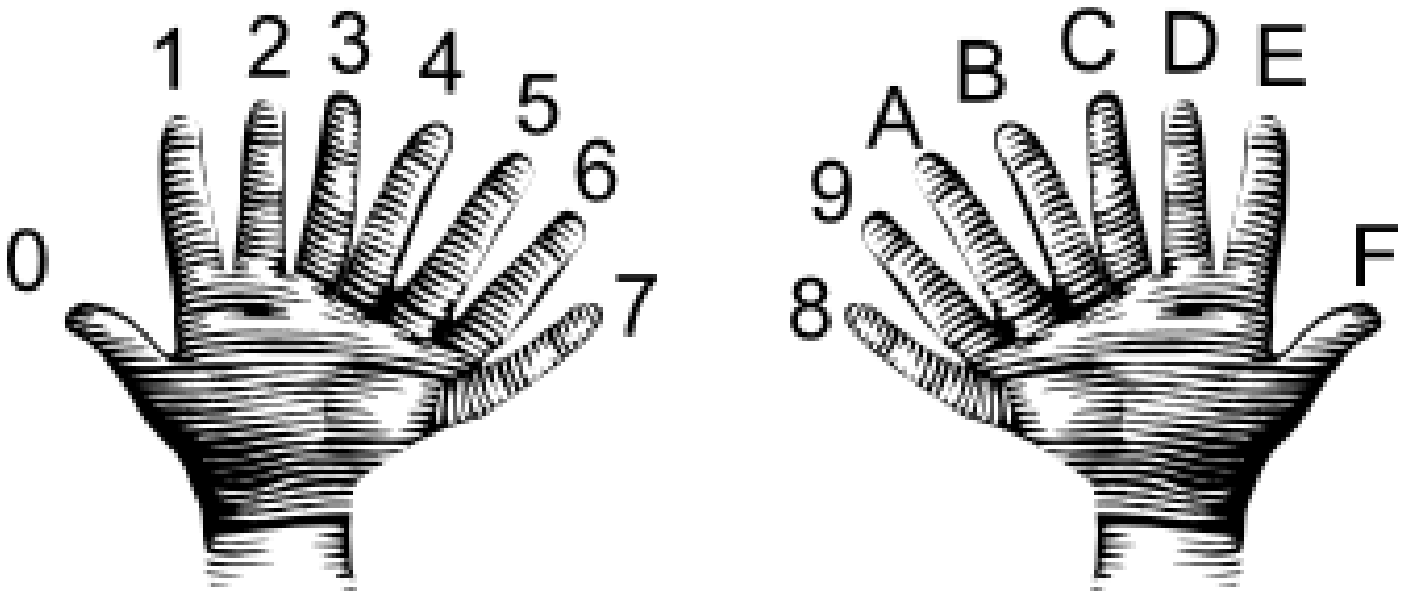


(from i-programmer.info)

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Decimal & Hexadecimal Numbers

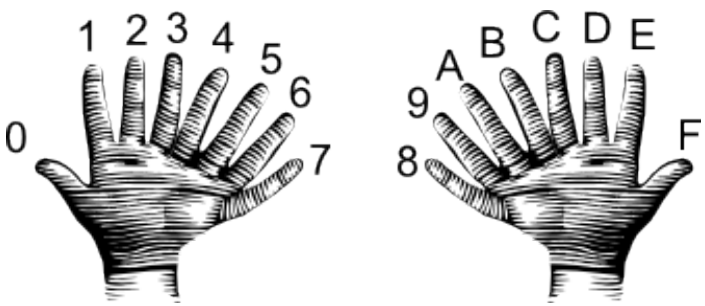
Counting with 16 digits:



(from i-programmer.info)

Hexadecimal

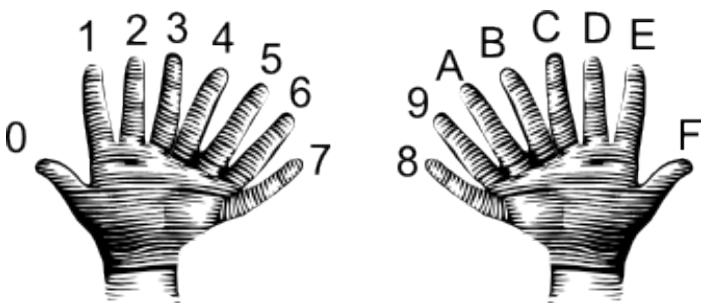
```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```



(from i-programmer.info)

Hexadecimal

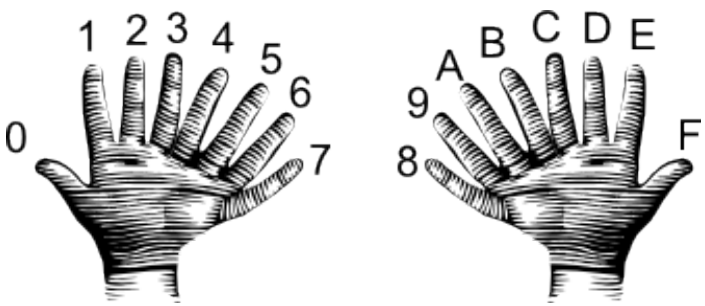
```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```



(from i-programmer.info)

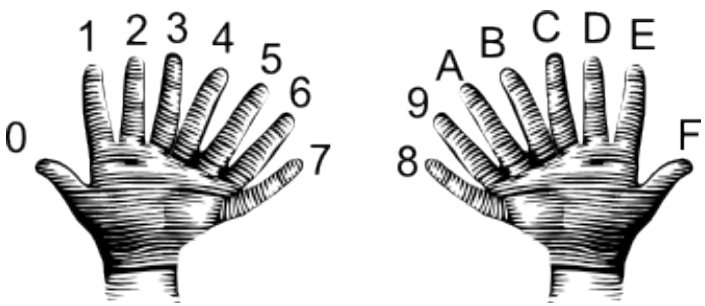
Hexadecimal

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
```



(from i-programmer.info)

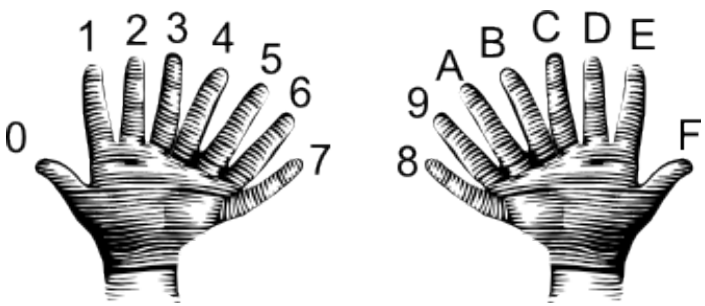
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
```

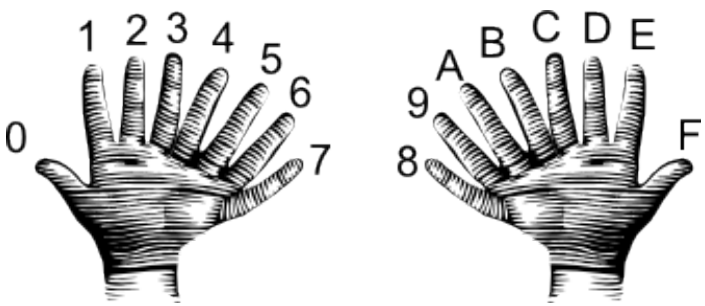
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
```

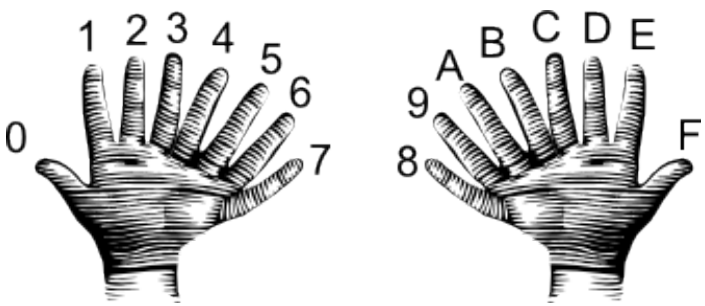
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
```

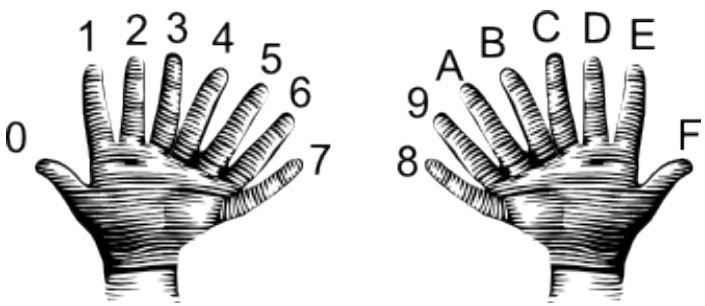
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
```

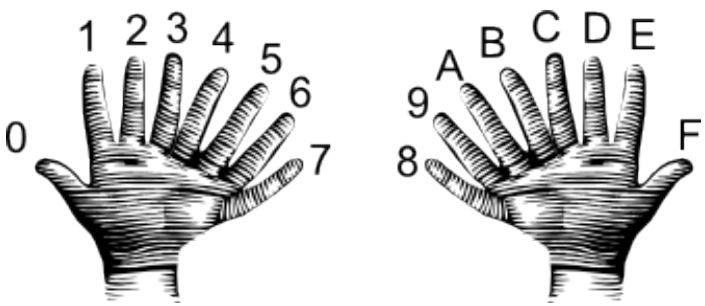
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
```

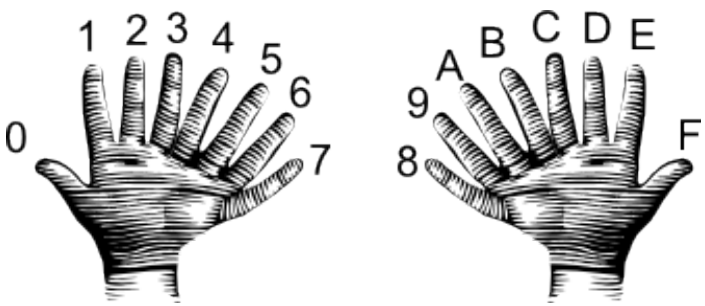
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
```

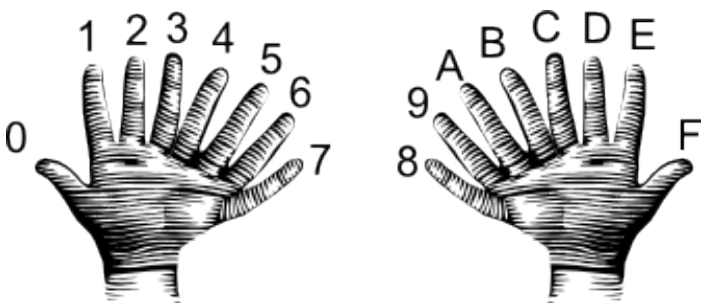
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
```

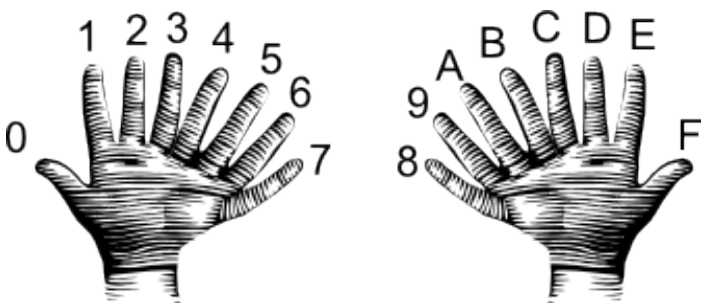
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
```

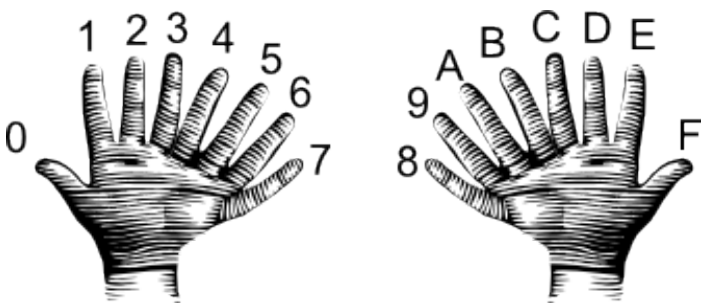
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
```

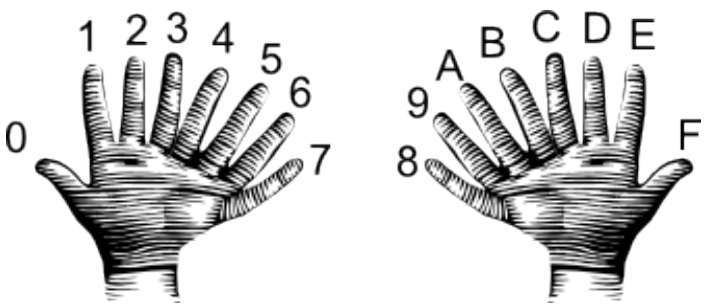
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
```

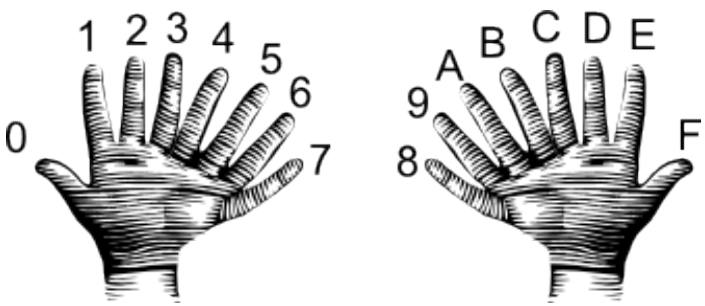
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
```

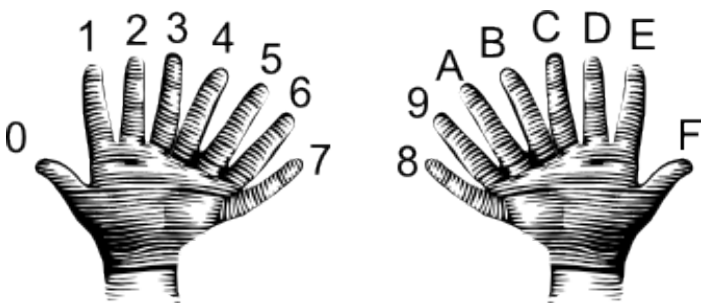
Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
```

Hexadecimal



(from i-programmer.info)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
```

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.
  - 8-bit colors: numbers from 0 to 255:
    e.g. `(0, 255, 0)` is no red, 100% green, and no blue.
  - Hexcodes (base-16 numbers):

Colors

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by numbers (RGB):
  - Fractions of each:
    e.g. `(1.0, 0, 0)` is 100% red, no green, and no blue.
  - 8-bit colors: numbers from 0 to 255:
    e.g. `(0, 255, 0)` is no red, 100% green, and no blue.
  - Hexcodes (base-16 numbers):
    e.g. `#0000FF` is no red, no green, and 100% blue.

In Pairs or Triples...

Some review and some novel challenges:

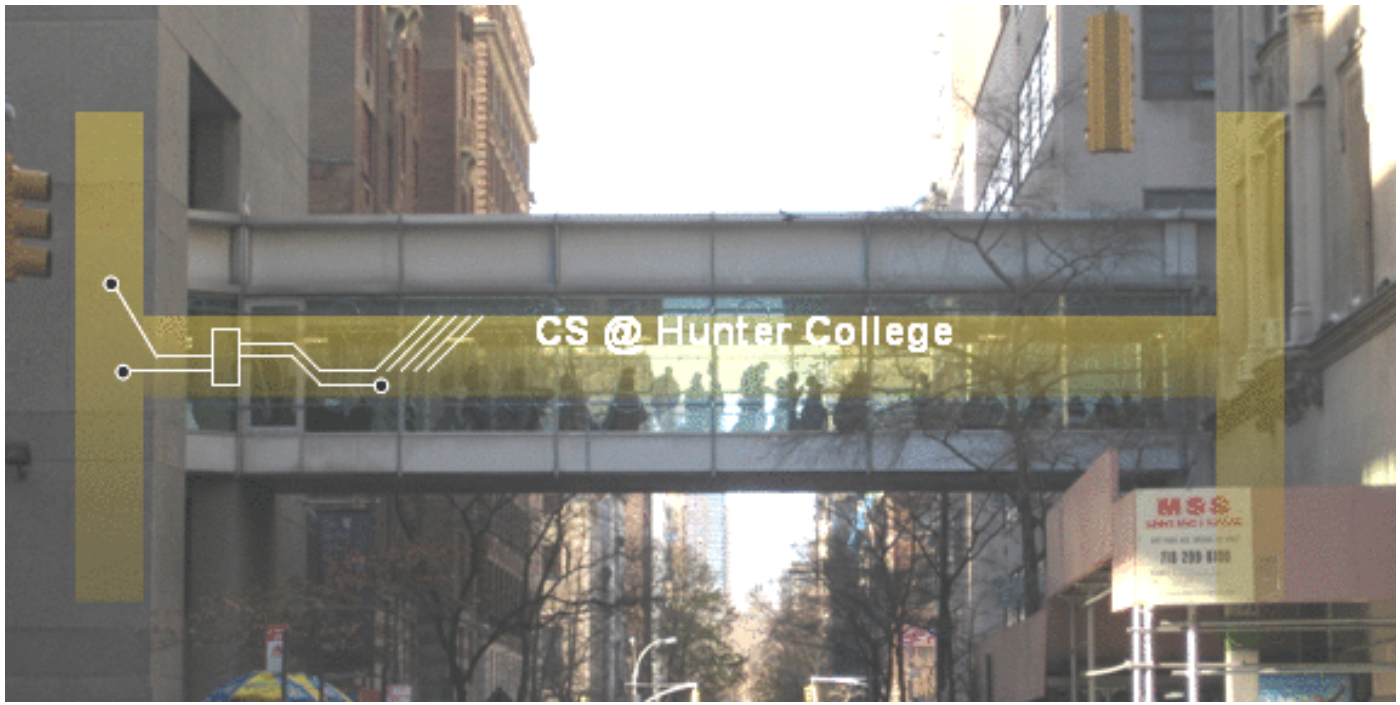
```
1  import turtle
2  teddy = turtle.Turtle()
3
4  names = ["violet", "purple", "indigo", "lavender"]
5  for c in names:
6    teddy.color(c)
7    teddy.left(60)
8    teddy.forward(40)
9    teddy.dot(10)
10
11 teddy.penup()
12 teddy.forward(100)
13 teddy.pendown()
14
15 hexNames = ["#FF00FF", "#990099", "#550055", "#111111"]
16 for c in hexNames:
17   teddy.color(c)
18   teddy.left(60)
19   teddy.forward(40)
20   teddy.dot(10)
```

Trinkets

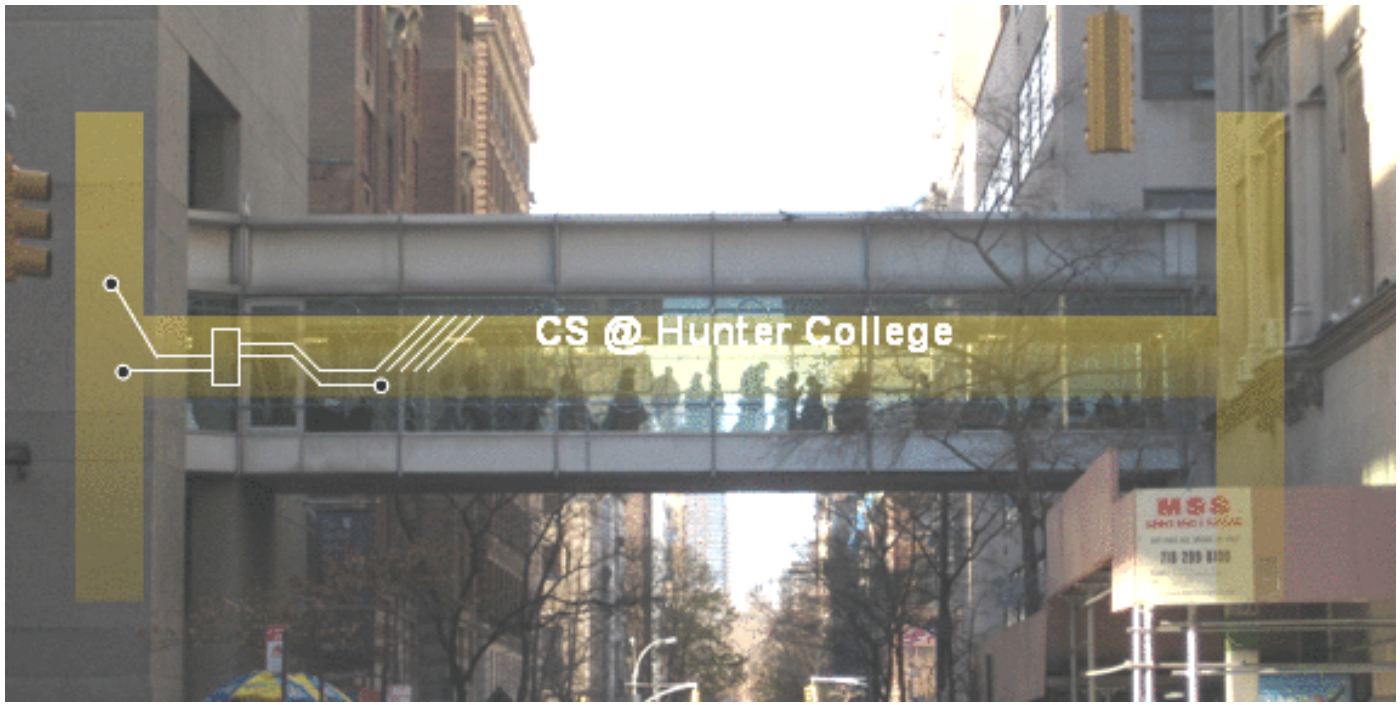
```
1  import turtle
2  teddy = turtle.Turtle()
3
4  names = ["violet", "purple", "indigo", "lavender"]
5  for c in names:
6    teddy.color(c)
7    teddy.left(60)
8    teddy.forward(40)
9    teddy.dot(10)
10
11 teddy.penup()
12 teddy.forward(100)
13 teddy.pendown()
14
15 hexNames = ["#FF00FF", "#990099", "#550055", "#111111"]
16 for c in hexNames:
17   teddy.color(c)
18   teddy.left(60)
19   teddy.forward(40)
20   teddy.dot(10)
```

(Demo with `trinkets`)

Images

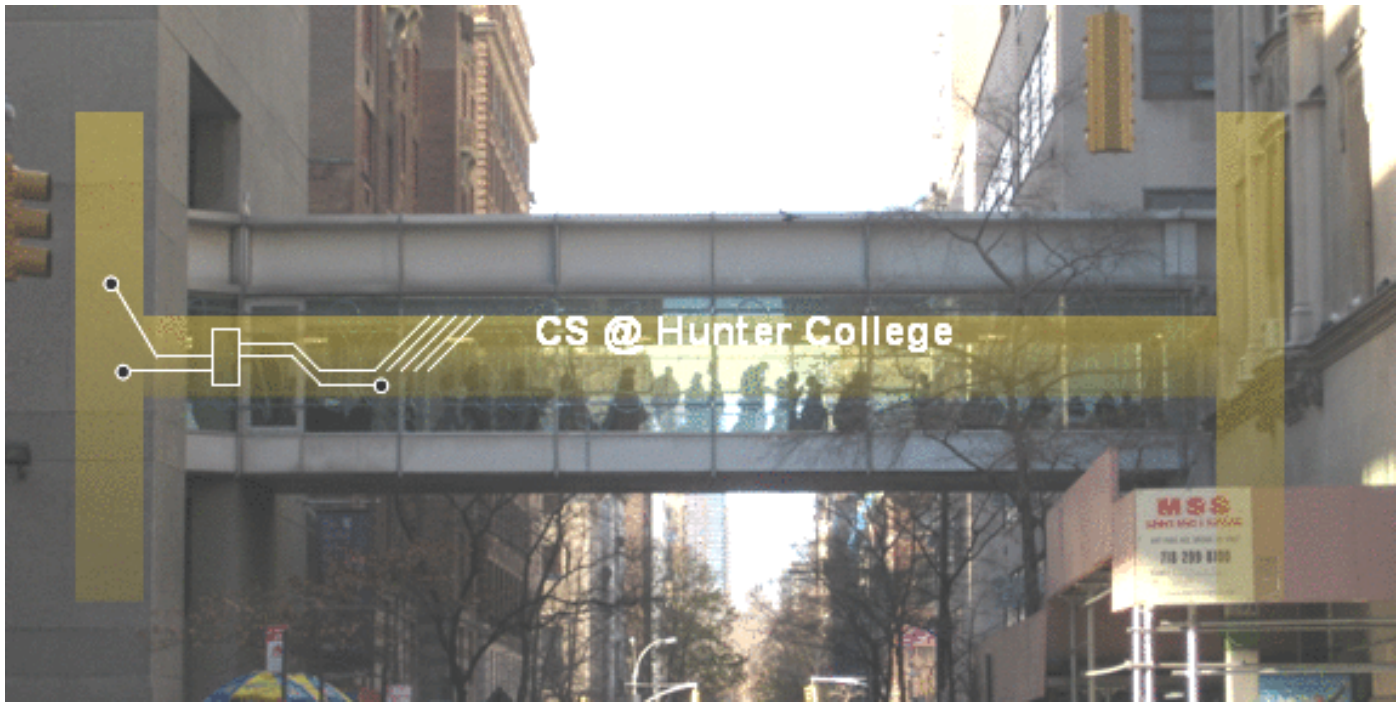


Images



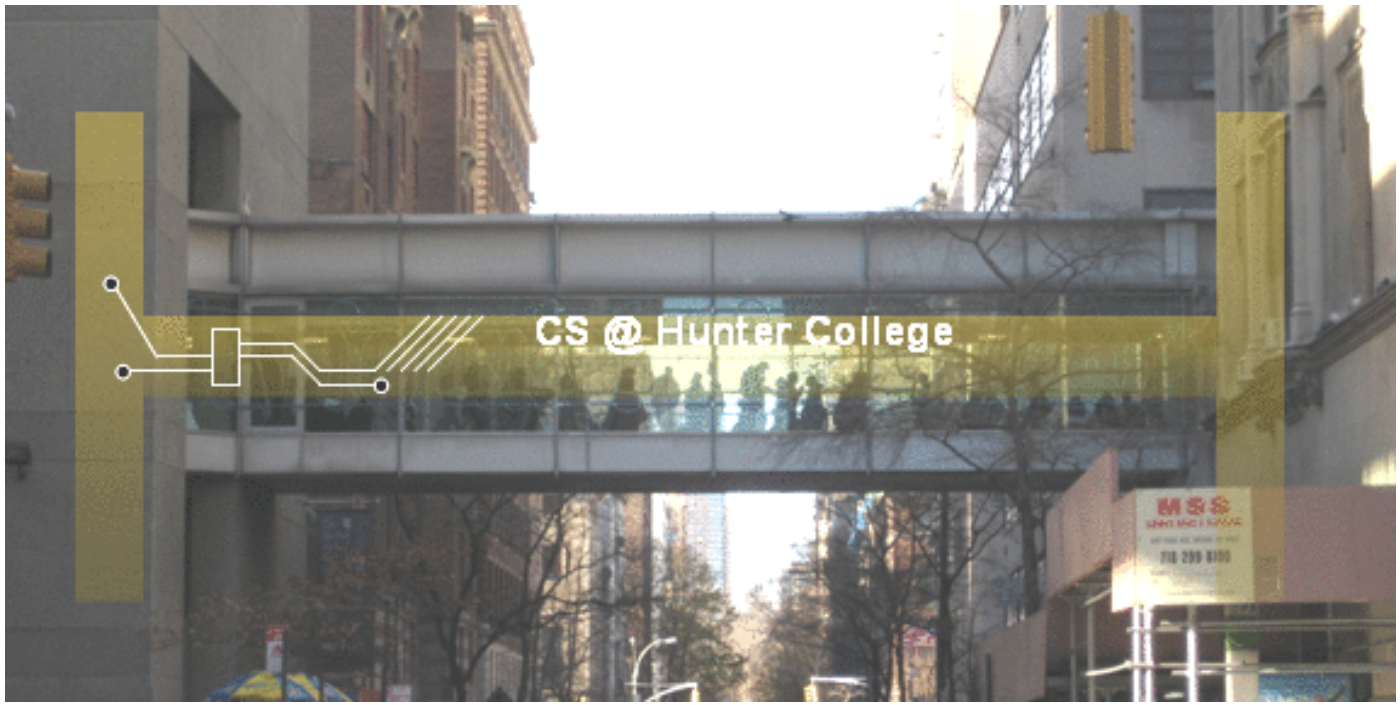
- We will use the standard portable network graphics (PNG) file format.

Images



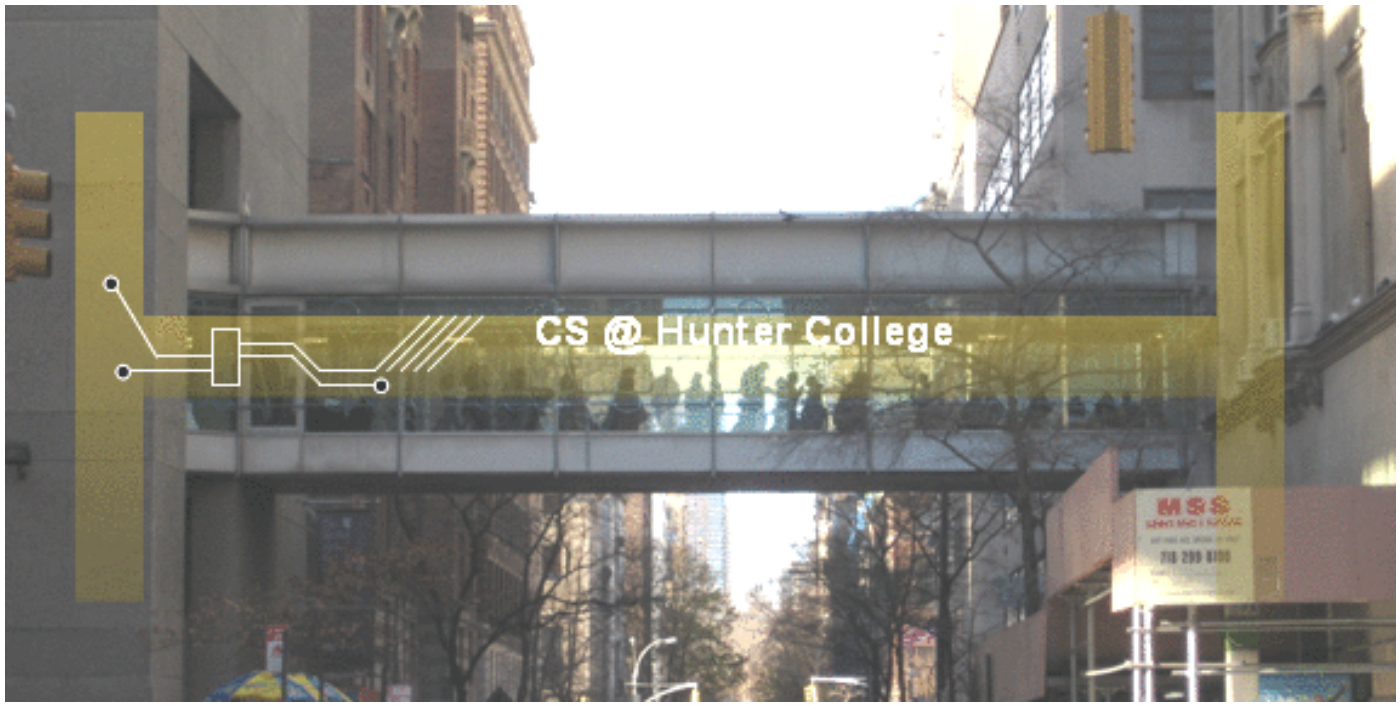
- We will use the standard portable network graphics (PNG) file format.
- Saves every picture element (or 'pixel')–

Images



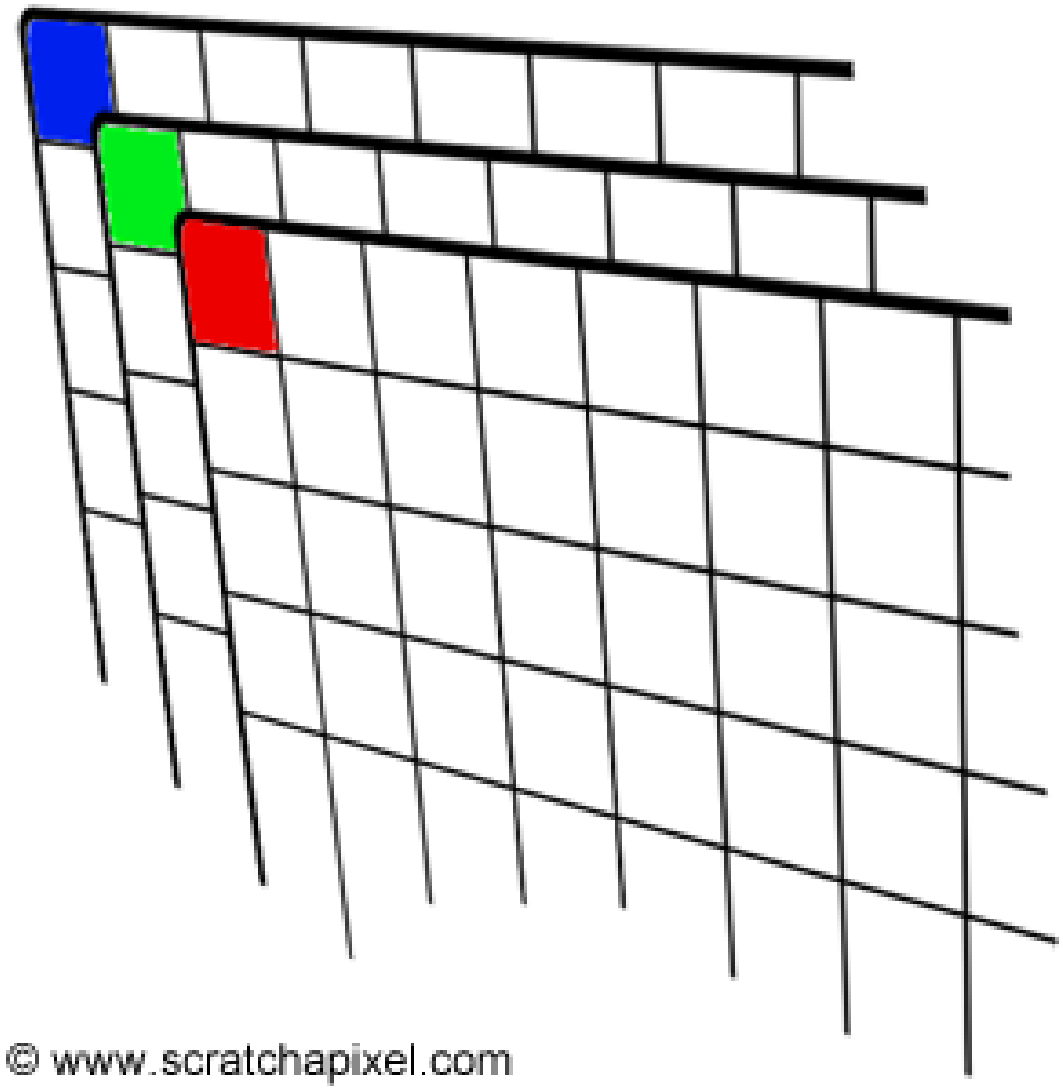
- We will use the standard portable network graphics (PNG) file format.
- Saves every picture element (or 'pixel')– often called a lossless format.

Images

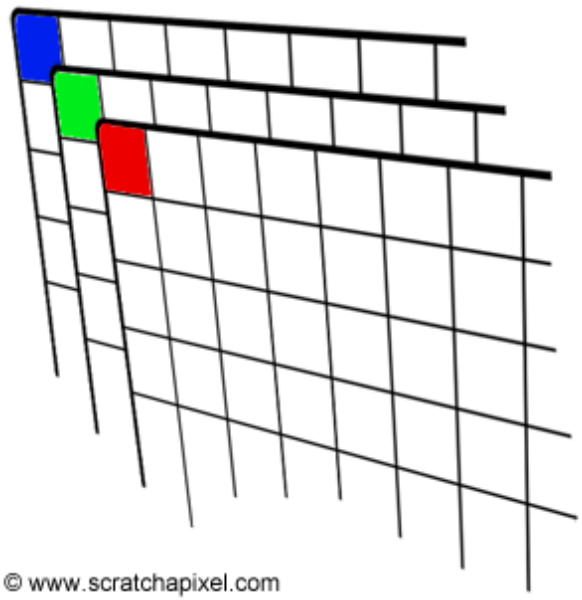


- We will use the standard portable network graphics (PNG) file format.
- Saves every picture element (or 'pixel')– often called a lossless format.
- Keeps track of the amount of red, blue, and green of each pixel.

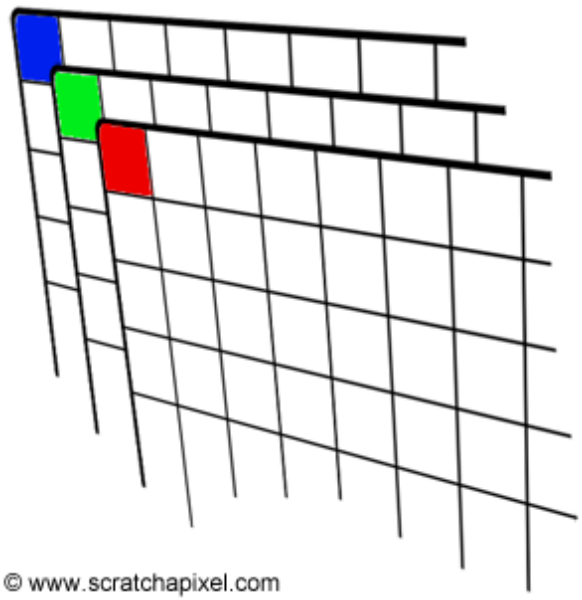
Images



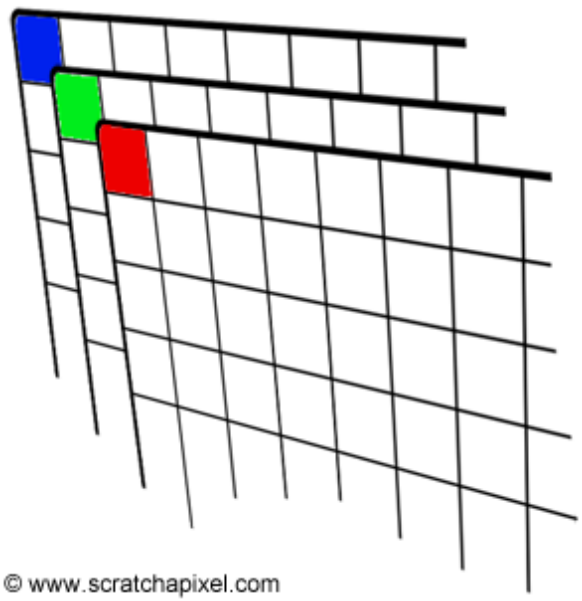
Images

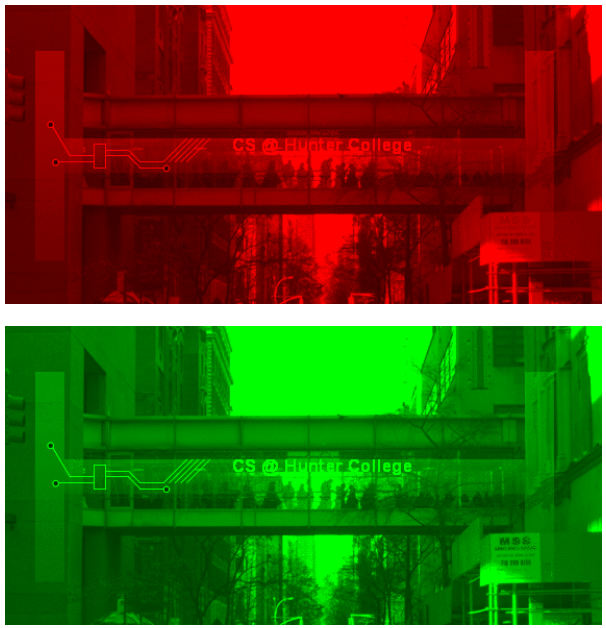


Images

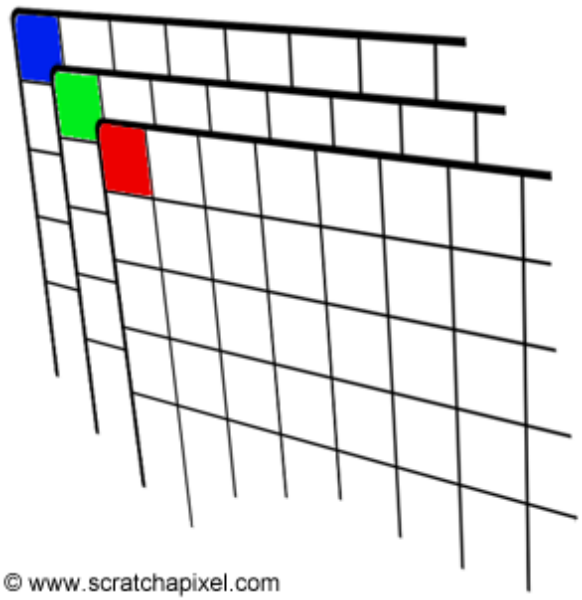


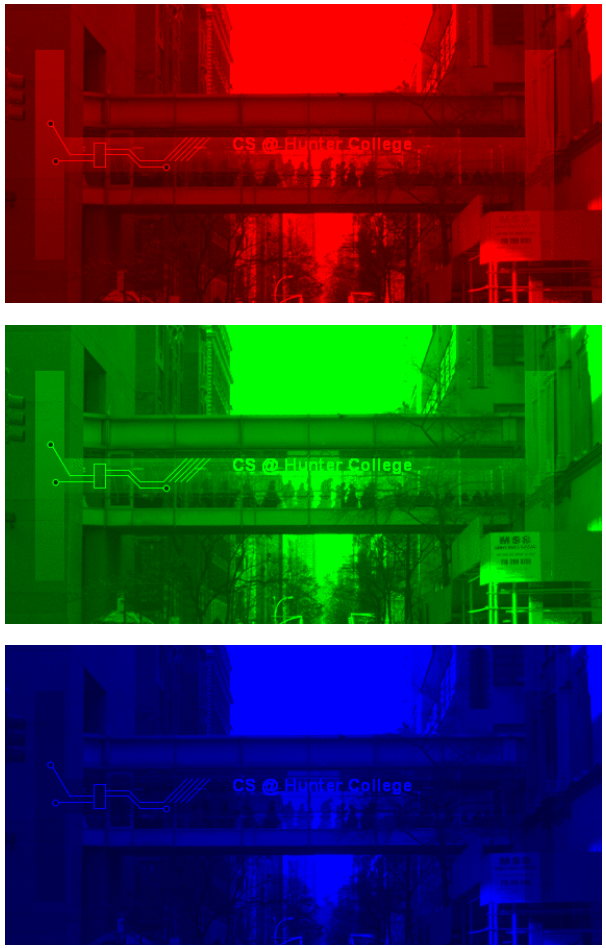
Images



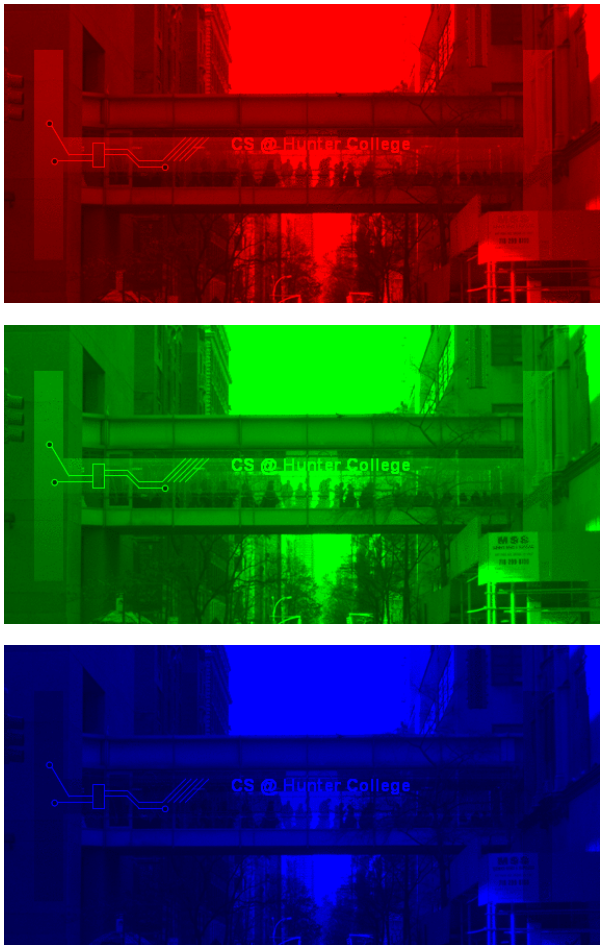


Images


© www.scratchapixel.com

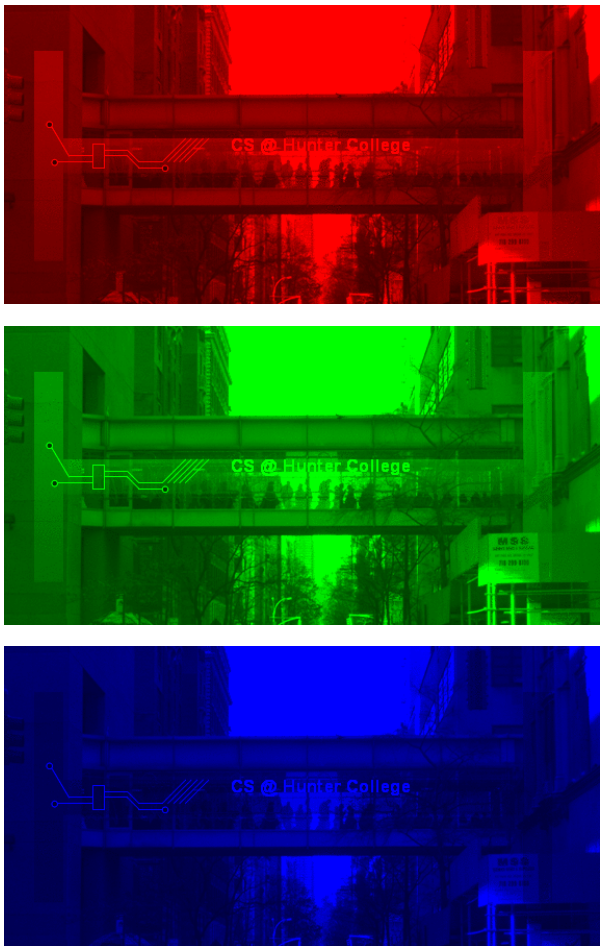

CS @ Hunter College
CS @ Hunter College
CS @ Hunter College

Useful Packages



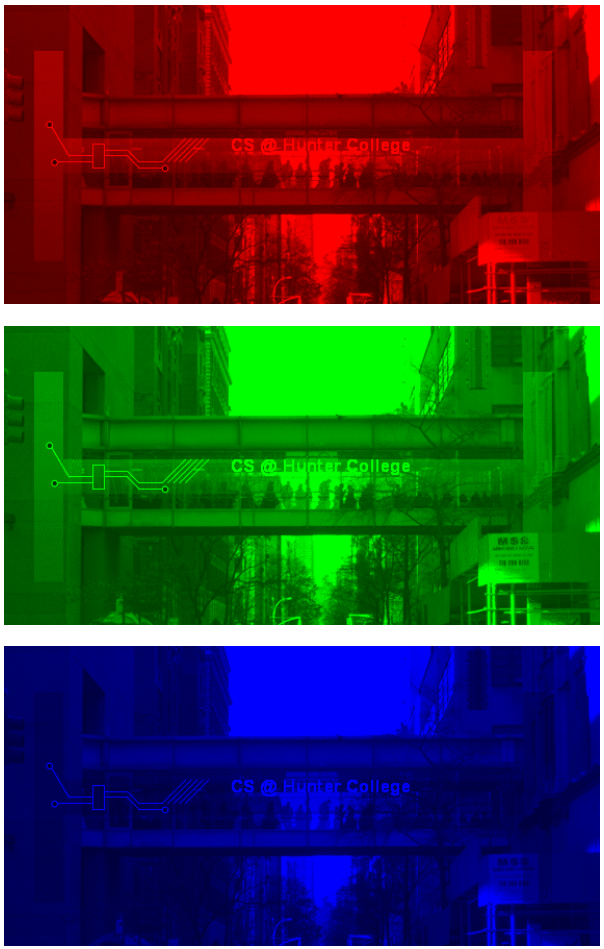
- We will use 2 useful packages for images:

Useful Packages



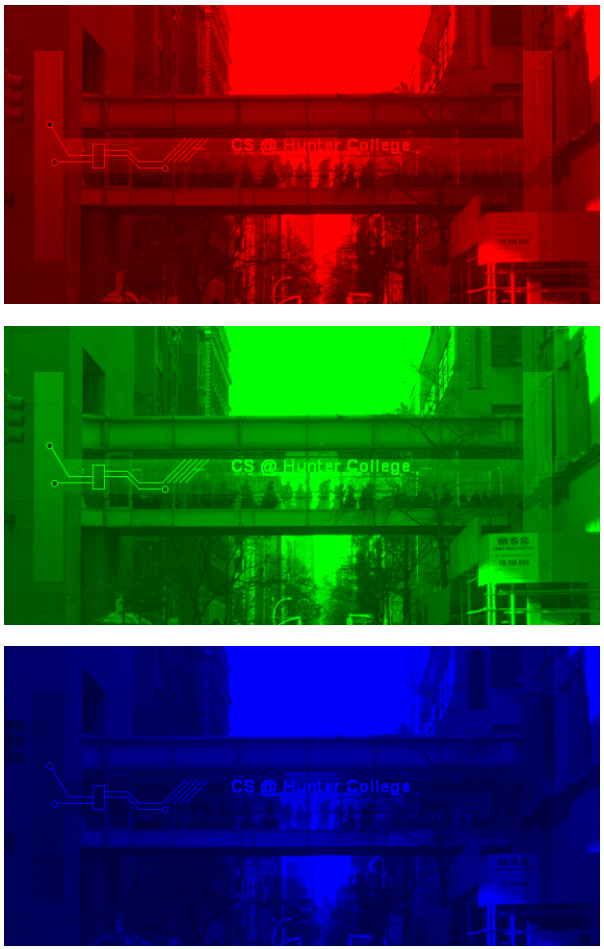
- We will use 2 useful packages for images:
  - `numpy`: numerical analysis package

Useful Packages



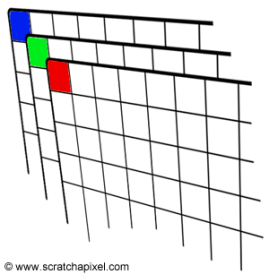
- We will use 2 useful packages for images:
  - `numpy`: numerical analysis package
  - `pyplot`: part of `matplotlib` for making graphs and plots

Useful Packages



- We will use 2 useful packages for images:
  - `numpy`: numerical analysis package
  - `pyplot`: part of `matplotlib` for making graphs and plots
- See lab notes for installing on your home machine.

Images with pyplot and numpy




```
#Import the packages for images and arrays:
import matplotlib.pyplot as plt
import numpy as np


img = plt.imread('csBridge.png')    #Read in image from csBridge.png
plt.imshow(img)                      #Load image into pyplot
plt.show()                           #Show the image (waits until close

img2 = img.copy()          #make a copy of our image
img2[:,:,1] = 0            #Set the green channel to 0
img2[:,:,2] = 0            #Set the blue channel to 0

plt.imshow(img2)           #Load our new image into pyplot
plt.show()                 #Show the image (waits until closed to conti

plt.imsave('reds.png', img2)  #Save the image we created to the file:
```

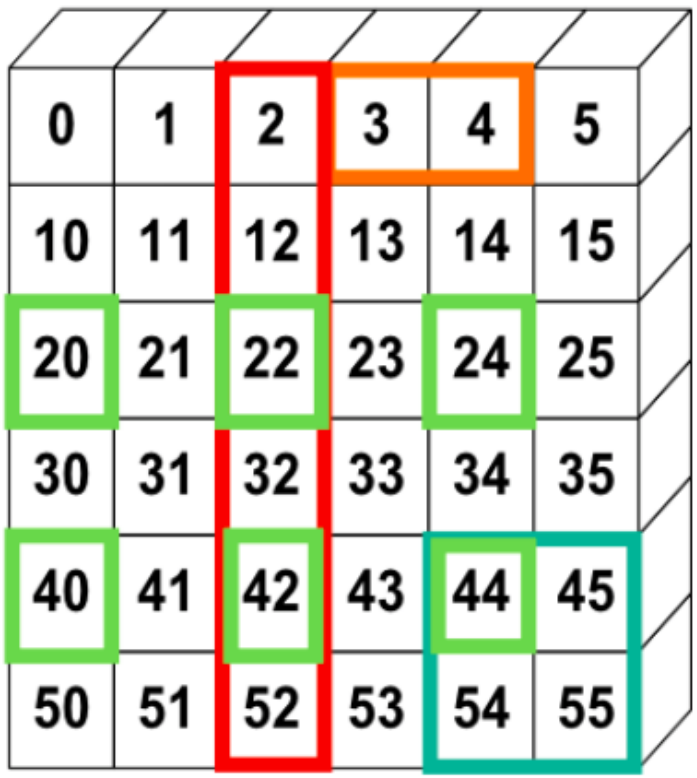
More on numpy arrays

```
>>> a[0,3:5]
array([3,4])

>>> a[4:,4:]
array([[44, 45],
       [54, 55]])

>>> a[:,2]
array([2,12,22,32,42,52])

>>> a[2::2,::2]
array([[20,22,24]
       [40,42,44]])
```



numpy tutorial

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

1. Design a 10 by 10 logo for Hunter College that contains a purple 'H'.

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

1. Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
2. Your logo should only contain the colors purple and white.

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

1. Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
2. Your logo should only contain the colors purple and white.
3. How can you make Python draw the logo?
   Write down a "To Do" list of things you need to do.

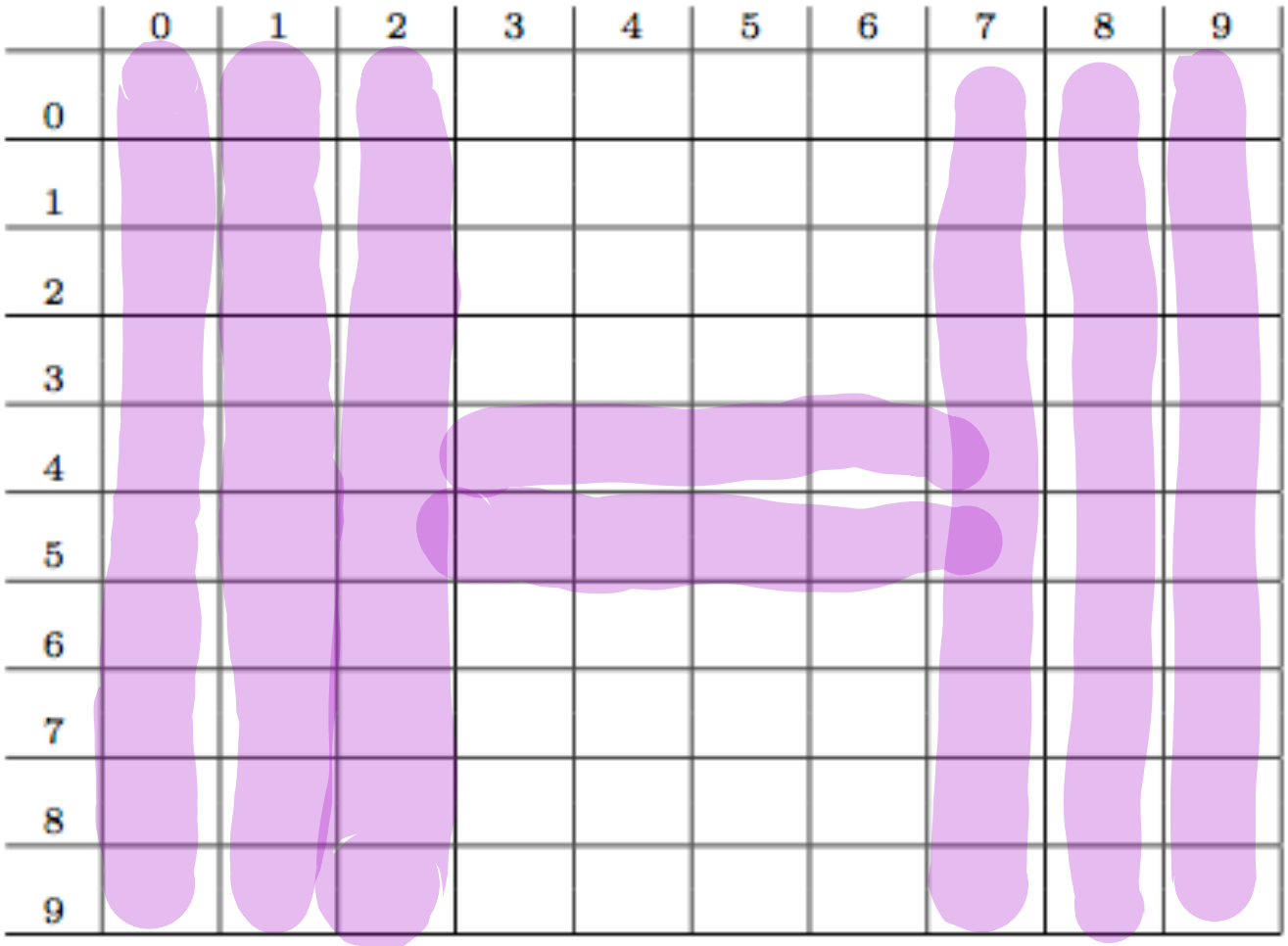
In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

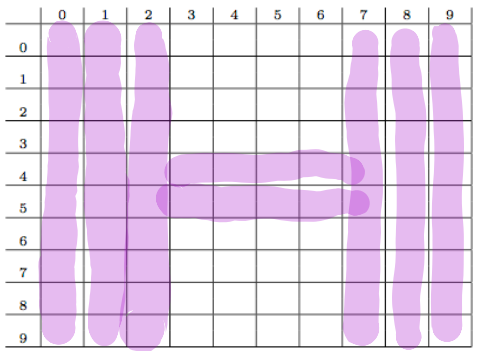
1. Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
2. Your logo should only contain the colors purple and white.
3. How can you make Python draw the logo?
   Write down a "To Do" list of things you need to do.
4. If time, refine your steps above into a Python program.

Design a Hunter Logo

One possible solution:

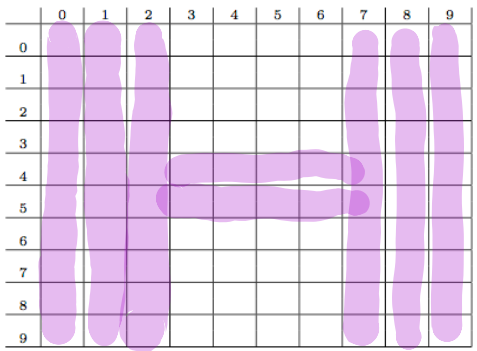


Design a Hunter Logo



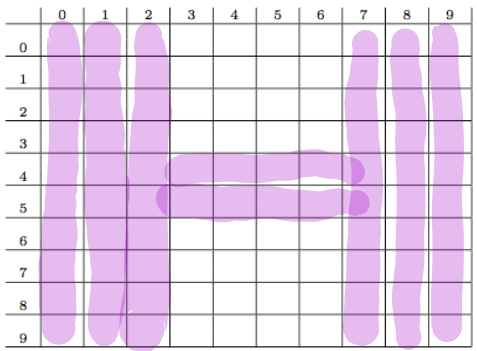
1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

Design a Hunter Logo



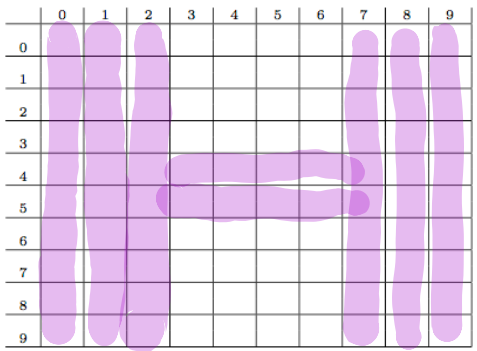
1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.
2. Set the 3 left columns to be purple.

Design a Hunter Logo



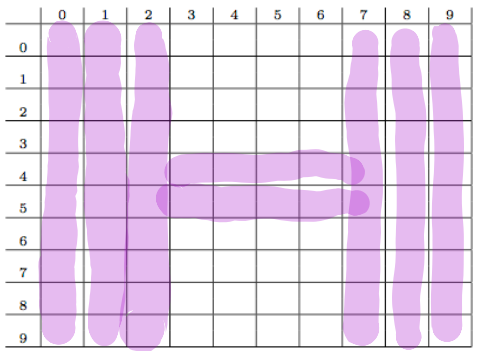
1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.
2. Set the 3 left columns to be purple.
3. Set the 3 right columns to be purple.

Design a Hunter Logo



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.
2. Set the 3 left columns to be purple.
3. Set the 3 right columns to be purple.
4. Set the middle 2 rows to be purple.

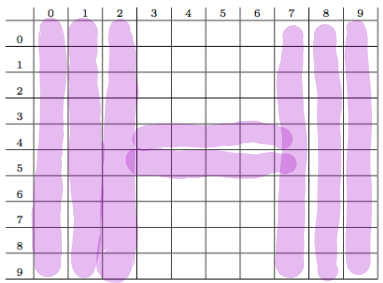
Design a Hunter Logo



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.
2. Set the 3 left columns to be purple.
3. Set the 3 right columns to be purple.
4. Set the middle 2 rows to be purple.
5. Save `logo` array to a file.

Translating the Design to Code

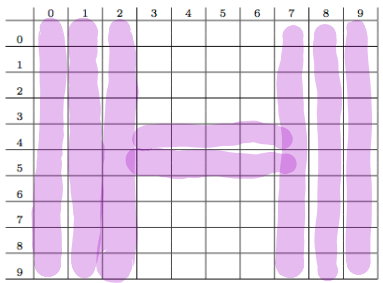
1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.



Translating the Design to Code

1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

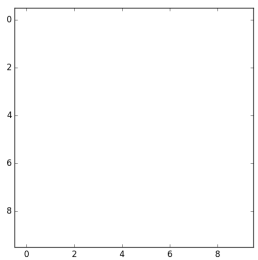
```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```



Translating the Design to Code

1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

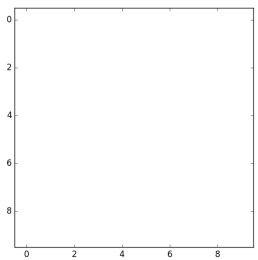


Translating the Design to Code

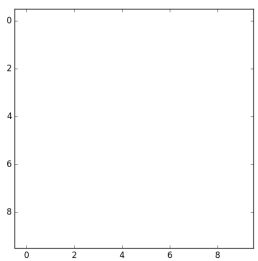
1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.



Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

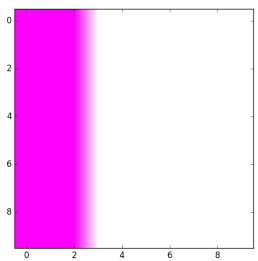
Translating the Design to Code

1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

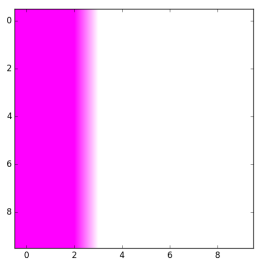
```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```



Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

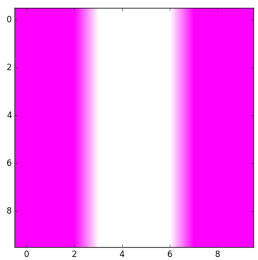
2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

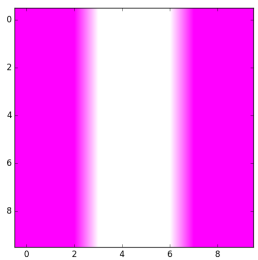
2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

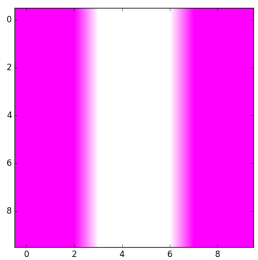
```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

4. Set the middle 2 rows to be purple.

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

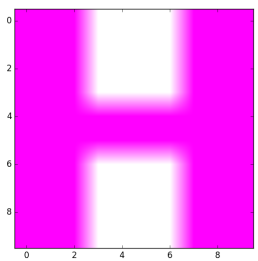
3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

4. Set the middle 2 rows to be purple.

```
logoImg[4:6,:,1] = 0 #Turn the green to 0 for middle rows
```

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

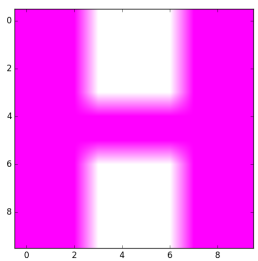
3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

4. Set the middle 2 rows to be purple.

```
logoImg[4:6,:,1] = 0 #Turn the green to 0 for middle rows
```

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

3. Set the 3 right columns to be purple.

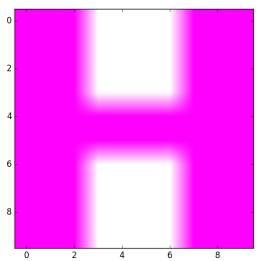
```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

4. Set the middle 2 rows to be purple.

```
logoImg[4:6,:,1] = 0 #Turn the green to 0 for middle rows
```

5. Save logo array to file.

Translating the Design to Code



1. Create a 10 by 10 array, `logo`, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np              #and for arrays (to hold images)
logoImg = np.ones((10,10,3))    #10x10 array with 3 sheets of 1's
```

2. Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:,:3,1] = 0 #Turn the green to 0 for first 3 columns
```

3. Set the 3 right columns to be purple.

```
logoImg[:,:-3,1] = 0 #Turn the green to 0 for last 3 columns
```

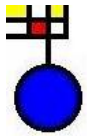
4. Set the middle 2 rows to be purple.

```
logoImg[4:6,:,1] = 0 #Turn the green to 0 for middle rows
```

5. Save logo array to file.

```
plt.imsave("logo.png", logoImg) #Save the image to logo.png
```

CS Survey: Prof. Sakas, Computational Linguistics

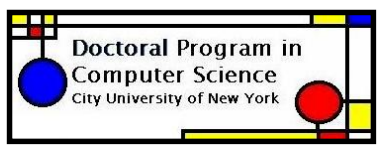


Language is Hard for Computers

Learning Language is Easy for my 3-year-old twins

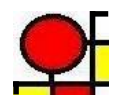
CSCI 12700 Guest Bullet Talk

William Gregory Sakas

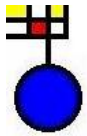


HUNTER
The City University of New York

M.A./Ph.D. Program in Linguistics
@ The City University of New York



CS Survey: Prof. Sakas, Computational Linguistics

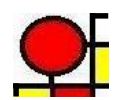


Language is Hard

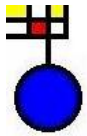
- *Buffalo buffalo, Buffalo buffalo buffalo, buffalo, Buffalo buffalo*

- *Someone shot the servant of the actress who was on the balcony.* **Who was on the balcony?**

- *Who do you think Mary kissed?*
- *Who do you think* ***that*** *Mary kissed?*
- *Who do you think bought a radio?*
- ****** *Who do you think* ***that*** *bought a radio?*



CS Survey: Prof. Sakas, Computational Linguistics



So how to explain language?

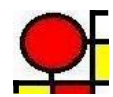
Treat Language as a **scientific field - like Physics.**

Example: *A* scientific principle about sentences:

Given <p> = [α [H β]],
where α = edge(*Spec's*) β then:
the head H of <p> is inert after the phase is completed, triggering no further grammatical operations.

Language is complex!!!
Understanding how language works is hard!!!

Unless you're 3.



CS Survey: Prof. Sakas, Computational Linguistics

Linguistic experts!

Lecture Slip

Linguistic experts!

Design a program that counts the number of plural nouns in a list of nouns. Think about:

- what the input is,
- what the output is, and
- how you can determine if a noun is plural.

Note: To simplify the problem, assume all plural nouns end in "s".

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Indexing and Slicing Lists

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Indexing and Slicing Lists
  - Colors

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Indexing and Slicing Lists
  - Colors
  - Hexadecimal Notation

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Indexing and Slicing Lists
  - Colors
  - Hexadecimal Notation
  - 2D Arrays & Image Files

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Indexing and Slicing Lists
  - Colors
  - Hexadecimal Notation
  - 2D Arrays & Image Files
- Pass your lecture slips to the end of the rows for the UTA's to collect.

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.

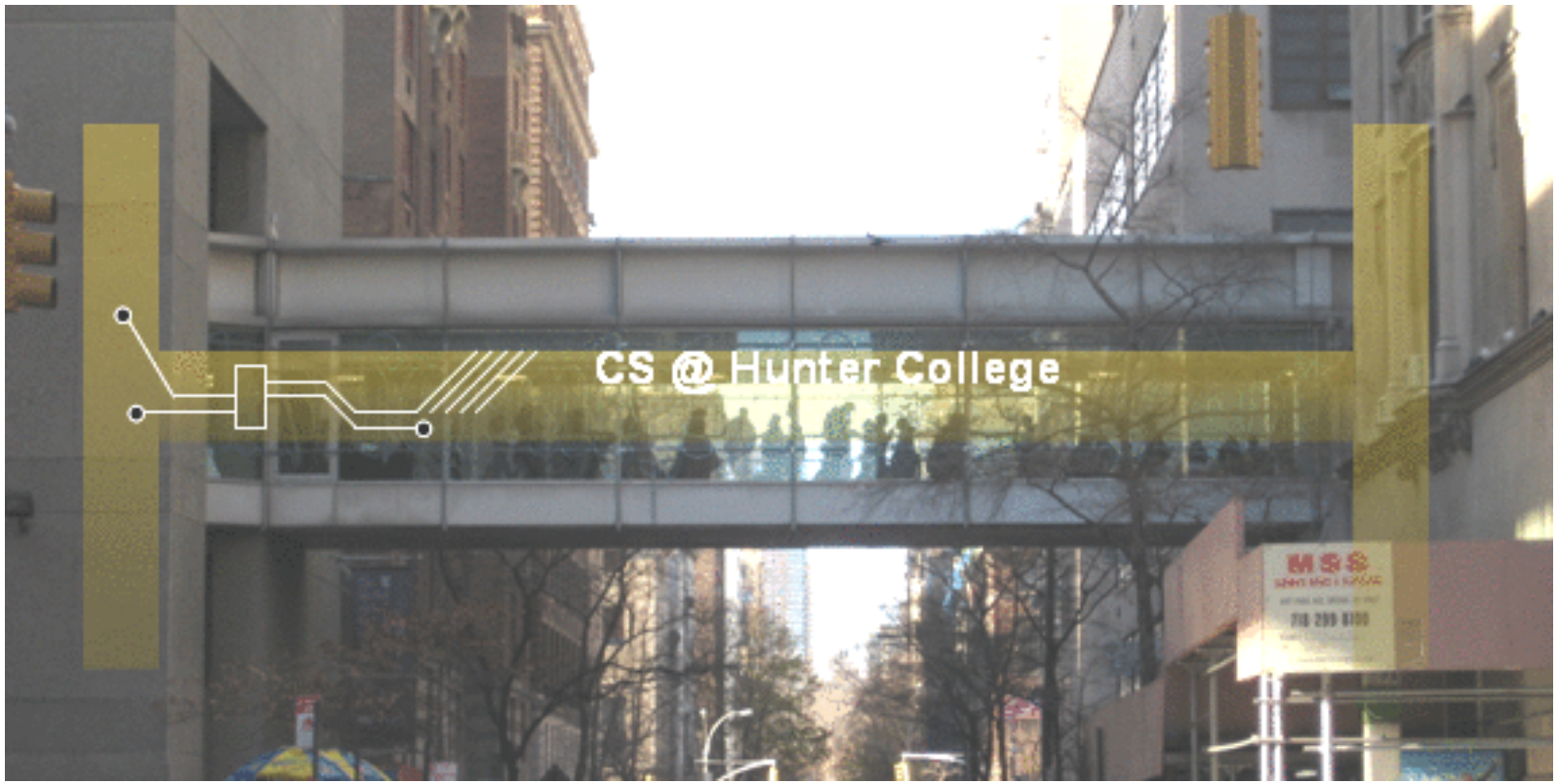
Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).

Practice Quiz & Final Questions

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).
- We're starting with Fall 2017, Version 2.

Writing Boards



- Return writing boards as you leave...