Frequently Asked Questions

From lecture slips & recitation sections.

When is the midterm?
There is no midterm. Instead there's 14 in-class quizzes.

When is the final?
Tuesday, 22 May, 9-11am.

Can I submit late homework?
No. Instead we drop the 5 lowest grades.

I missed class. Do you need documentation?
No. Missing lecture slip & quiz grades are replaced by your final exam score.

If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.

Why do I have to work in groups?
It's great practice to explain technical work to others.

Can I work ahead?
Yes! All programs are available, on gradescope, 4 weeks before the deadline.

You said "when you take second semester..." I just took this class for Pathways...
This is Pathways, but we hope that you will be a CS major/minor.
We also hope: "Get your education don't forget whence you came..."
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?

No midterm. Instead, there are 14 in-class quizzes.

- When is the final?

Tuesday, 22 May, 9-11am.

- Can I submit late homework?

No. Instead, we drop the 5 lowest grades.

- I missed class. Do you need documentation?

No. Missing lecture slip & quiz grades are replaced by your final exam score.

- If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.

- Why do I have to work in groups?

It's great practice to explain technical work to others.

- Can I work ahead?

Yes! All programs are available, on gradescope, 4 weeks before the deadline.

- You said "when you take second semester..." I just took this class for Pathways...

This is Pathways, but we hope that you will be a CS major/minor.

We also hope: "Get your education don't forget whence you came..."
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  
  *There is no midterm. Instead there’s 14 in-class quizzes.*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there's 14 in-class quizzes.*

- When is the final?
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

---

CSci 127 (Hunter)  
Lecture 2  
7 February 2018  
2 / 26
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?

---

CSci 127 (Hunter)  Lecture 2  7 February 2018  2 / 26
Frequently Asked Questions
From lecture slips & recitation sections.

- When is the midterm?
  
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  
  *No. Instead we drop the 5 lowest grades.*
Frequently Asked Questions

From lecture slips & recitation sections.

• When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

• When is the final?
  *Tuesday, 22 May, 9-11am.*

• Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

• I missed class. Do you need documentation?

  *Missing lecture slip & quiz grades are replaced by your final exam score.*

  *If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

• Why do I have to work in groups?
  *It’s great practice to explain technical work to others.*

• Can I work ahead?
  *Yes! All programs are available, on gradescope, 4 weeks before the deadline.*

• You said “when you take second semester...” I just took this class for Pathways...
  *This is Pathways, but we hope that you will be a CS major/minor.*
  *We also hope: “Get your education don’t forget whence you came...”*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there's 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

- Why do I have to work in groups?
Frequently Asked Questions

From lecture slips & recitation sections.

- **When is the midterm?**
  
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- **When is the final?**
  
  *Tuesday, 22 May, 9-11am.*

- **Can I submit late homework?**
  
  *No. Instead we drop the 5 lowest grades.*

- **I missed class. Do you need documentation?**
  
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

- **Why do I have to work in groups?**
  
  *It’s great practice to explain technical work to others.*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss $\geq 3$ weeks ($>20\%$), see us about taking this in a future term.*

- Why do I have to work in groups?
  *It’s great practice to explain technical work to others.*

- Can I work ahead?
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss \( \geq 3 \) weeks (> 20%), see us about taking this in a future term.*

- Why do I have to work in groups?
  *It’s great practice to explain technical work to others.*

- Can I work ahead?
  *Yes! All programs are available, on gradescope, 4 weeks before the deadline.*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

- Why do I have to work in groups?
  *It’s great practice to explain technical work to others.*

- Can I work ahead?
  *Yes! All programs are available, on gradescope, 4 weeks before the deadline.*

- You said “when you take second semester...” I just took this class for Pathways...
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  
  *No. Missing lecture slip & quiz grades are replaced by your final exam score. If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

- Why do I have to work in groups?
  
  *It’s great practice to explain technical work to others.*

- Can I work ahead?
  
  *Yes! All programs are available, on gradescope, 4 weeks before the deadline.*

- You said “when you take second semester...” I just took this class for Pathways...
  
  *This is Pathways, but we hope that you will be a CS major/minor.*
Frequently Asked Questions

From lecture slips & recitation sections.

- When is the midterm?
  *There is no midterm. Instead there’s 14 in-class quizzes.*

- When is the final?
  *Tuesday, 22 May, 9-11am.*

- Can I submit late homework?
  *No. Instead we drop the 5 lowest grades.*

- I missed class. Do you need documentation?
  *No. Missing lecture slip & quiz grades are replaced by your final exam score.*
  *If you will miss ≥ 3 weeks (> 20%), see us about taking this in a future term.*

- Why do I have to work in groups?
  *It’s great practice to explain technical work to others.*

- Can I work ahead?
  *Yes! All programs are available, on gradescope, 4 weeks before the deadline.*

- You said “when you take second semester...” I just took this class for Pathways...
  *This is Pathways, but we hope that you will be a CS major/minor.*
  *We also hope: “Get your education don’t forget whence you came...”*
Today’s Topics

- For-loops
- `range()`
- Variables: ints and strings
- Lists
In Pairs or Triples...

Some review and some novel challenges:

```python
#Predict what will be printed:
for i in range(4):
    print('The world turned upside down')
for j in [0,1,2,3,4,5]:
    print(j)
for count in range(6):
    print(count)
for color in ['red', 'green', 'blue']:
    print(color)
for i in range(2):
    for j in range(2):
        print('Look around,')
print('How lucky we are to be alive!')
```
Python Tutor

```python
#Predict what will be printed:
for i in range(4):
    print('The world turned upside down')
for j in [0,1,2,3,4,5]:
    print(j)
for count in range(6):
    print(count)
for color in ['red', 'green', 'blue']:
    print(color) |
for i in range(2):
    for j in range(2):
        print('Look around,')
print('How lucky we are to be alive!')
```
Variables

- A **variable** is a reserved memory location for storing a value.
Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
  - **int**: integer or whole numbers
  - **float**: floating point or real numbers
  - **string**: sequence of characters
  - **list**: a sequence of items e.g. `[3, 1, 4, 5, 9]` or `['violet', 'purple', 'indigo']`
  - **class variables**: for complex objects, like turtles.
Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
  - **int**: integer or whole numbers
  - **float**: floating point or real numbers
Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
  - **int**: integer or whole numbers
  - **float**: floating point or real numbers
  - **string**: sequence of characters
A variable is a reserved memory location for storing a value.

Different kinds, or types, of values need different amounts of space:

- **int**: integer or whole numbers
- **float**: floating point or real numbers
- **string**: sequence of characters
- **list**: a sequence of items
A variable is a reserved memory location for storing a value.

Different kinds, or types, of values need different amounts of space:

- **int**: integer or whole numbers
- **float**: floating point or real numbers
- **string**: sequence of characters
- **list**: a sequence of items

E.g. [3, 1, 4, 5, 9] or ['violet', 'purple', 'indigo']
Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
  - **int**: integer or whole numbers
  - **float**: floating point or real numbers
  - **string**: sequence of characters
  - **list**: a sequence of items
    e.g. [3, 1, 4, 5, 9] or ['violet','purple','indigo']
  - **class variables**: for complex objects, like turtles.
There’s some rules about valid names for variables.
Variable Names

- There's some rules about valid names for variables.
- Can use the underscore ('_') upper and lower case letters.
Variable Names

- There’s some rules about valid names for variables.
- Can use the underscore (‘_’), upper and lower case letters.
- Can also use numbers, just can’t start a name with a number.
Variable Names

- There’s some rules about valid names for variables.
- Can use the underscore (‘_’), upper and lower case letters.
- Can also use numbers, just can’t start a name with a number.
- Can’t use symbols (like ’+’ or ’*’) since used for arithmetic.
Variable Names

- There’s some rules about valid names for variables.
- Can use the underscore (’_’), upper and lower case letters.
- Can also use numbers, just can’t start a name with a number.
- Can’t use symbols (like ’+’ or ’*’) since used for arithmetic.
- Can’t use some words that Python has reserved for itself (like `for`).
  (List of reserved words in *Think CS*, §2.5.)
for-loop

for i in list:
  statement1
  statement2
  statement3

How to Think Like CS, §4.5
for-loop

for i in list:
    statement1
    statement2
    statement3

where list is a list of items:
  • stated explicitly (e.g. [1, 2, 3]) or
  • generated by a function, e.g. range().

*How to Think Like CS, §4.5*
In Pairs or Triples...

Some review and some novel challenges:

```python
#Predict what will be printed:

for num in [2,4,6,8,10]:
    print(num)

sum = 0
for x in range(0,12,2):
    print(x)
    sum = sum + x

print(x)

for c in "ABCD":
    print(c)
```
Python Tutor

(Demo with pythonTutor)
range()

Simplest version:
range()

Simplest version:

- range(stop)
range()

Simplest version:

- `range(stop)`
- Produces a list: [0,1,2,3,...,stop-1]
range()

Simplest version:

- range(stop)
- Produces a list: [0,1,2,3,...,stop-1]
- For example, if you want the list [0,1,2,3,...,100], you would write:
range()  

Simplest version:

- \texttt{range(stop)}
- Produces a list: \[0, 1, 2, 3, \ldots, \text{stop}-1\]
- For example, if you want the list \([0, 1, 2, 3, \ldots, 100]\), you would write:

\texttt{range(101)}
range()

Simplest version:

- `range(stop)`
- Produces a list: `[0,1,2,3,...,stop-1]`
- For example, if you want the list `[0,1,2,3,...,100]`, you would write:
range()

Simplest version:

- `range(stop)`
- Produces a list: `[0,1,2,3,...,stop-1]`
- For example, if you want the list `[0,1,2,3,...,100]`, you would write:

  `range(101)`
range()

What if you wanted to start somewhere else:

```python
range(start, stop)
```

Produces a list:

```
[start, start+1, ..., stop-1]
```

For example, if you want the list `[10, 11, ..., 20]` you would write:

```python
range(10, 21)
```
range()

What if you wanted to start somewhere else:

- \texttt{range(start, stop)}

For example, if you want the list \[10, 11, \ldots, 20\] you would write: \texttt{range(10, 21)}
range()

What if you wanted to start somewhere else:

- range(start, stop)
- Produces a list: [start, start+1, ..., stop-1]
range()

What if you wanted to start somewhere else:

- `range(start, stop)`
- Produces a list:
  
  \[ \text{[start, start+1, \ldots, stop-1]} \]
- For example, if you want the list \[10, 11, \ldots, 20\],
  
you would write:
range()

What if you wanted to start somewhere else:

- range(start, stop)
- Produces a list: [start, start+1,...,stop-1]
- For example, if you want the list [10,11,...,20] you would write:

  range(10,21)
range()

What if you wanted to start somewhere else:

- `range(start, stop)`
- Produces a list: `[start, start+1, ..., stop-1]`
- For example, if you want the list `[10, 11, ..., 20]` you would write:

  `range(10, 21)`
range()

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

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

- `range(start, stop, step)`

```python
range(5, 51, 5)
```
range()

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()

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 list `[5,10,...,50]` you would write:
range()

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 list \([5,10,...,50]\)
  
  you would write:

  `range(5,51,5)`
range()

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 list `[5, 10, ..., 50]`
  you would write:
range()

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

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

  \[\text{range(5, 51, 5)}\]
In summary: range()

The three versions:
In summary: `range()`

The three versions:
- `range(stop)`
In summary: range()

The three versions:
  * range(stop)
  * range(start, stop)
In summary: range()

The three versions:

- range(stop)
- range(start, stop)
- range(start, stop, step)
Standardized Code for Characters
Standardized Code for Characters
Standardized Code for Characters
(New version called: Unicode).

ASCII TABLE

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hex</th>
<th>Char</th>
<th>Decimal</th>
<th>Hex</th>
<th>Char</th>
<th>Decimal</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>[NULL]</td>
<td>32</td>
<td>20</td>
<td>[SPACE]</td>
<td>64</td>
<td>40</td>
<td>@</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>[START OF HEADING]</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>[START OF TEXT]</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>[END OF TEXT]</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>[END OF TRANSMISSION]</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>44</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>[ENQUIRY]</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>E</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>[ACKNOWLEDGE]</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>46</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>[BEL]</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>47</td>
<td>G</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>[BACKSPACE]</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>48</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>[HORIZONTAL TAB]</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>49</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>[LINE FEED]</td>
<td>42</td>
<td>2A</td>
<td>*</td>
<td>74</td>
<td>4A</td>
<td>J</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>[VERTICAL TAB]</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>75</td>
<td>4B</td>
<td>K</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>[FORM FEED]</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>76</td>
<td>4C</td>
<td>L</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>[CARRIAGE RETURN]</td>
<td>45</td>
<td>2D</td>
<td>-</td>
<td>77</td>
<td>4D</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>[SHIFT OUT]</td>
<td>46</td>
<td>2E</td>
<td>.</td>
<td>78</td>
<td>4E</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>[SHIFT IN]</td>
<td>47</td>
<td>2F</td>
<td>/</td>
<td>79</td>
<td>4F</td>
<td>O</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>[DATA LINK ESCAPE]</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>P</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>[DEVICE CONTROL 1]</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>Q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>[DEVICE CONTROL 2]</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>R</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>[DEVICE CONTROL 3]</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>S</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>[DEVICE CONTROL 4]</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>T</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>[NEGATIVE ACKNOWLEDGE]</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>U</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>[SYNCHRONOUS IDLE]</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>V</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>[ENG OF TRANS, BLOCK]</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>W</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>[CANCEL]</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>X</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>[END OF MEDIUM]</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>Y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>[SUBSTITUTE]</td>
<td>58</td>
<td>3A</td>
<td>:</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>[ESCAPE]</td>
<td>59</td>
<td>3B</td>
<td>;</td>
<td>91</td>
<td>5B</td>
<td>[</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>[FILE SEPARATOR]</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
<td>92</td>
<td>5C</td>
<td>\</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>[GROUP SEPARATOR]</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>93</td>
<td>5D</td>
<td>]</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>[RECORD SEPARATOR]</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
<td>94</td>
<td>5E</td>
<td>^</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>[UNIT SEPARATOR]</td>
<td>63</td>
<td>3F</td>
<td>?</td>
<td>95</td>
<td>5F</td>
<td>`</td>
</tr>
</tbody>
</table>

7 February 2018
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)

- \texttt{ord(c)}: returns Unicode (ASCII) of the character.
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)

- `ord(c)`: returns Unicode (ASCII) of the character.
- Example: `ord('a')` returns 97.
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)

- \texttt{ord(c)}: returns Unicode (ASCII) of the character.
- Example: \texttt{ord(’a’)} returns 97.
- \texttt{chr(x)}: returns the character whose Unicode is \texttt{x}.
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)

- `ord(c)`: returns Unicode (ASCII) of the character.
  - Example: `ord(’a’)` returns 97.
- `chr(x)`: returns the character whose Unicode is `x`.
  - Example: `chr(97)` returns ’a’.
Converting from Character to Code:

(There is an ASCII table on the back of today’s lecture slip.)

- `ord(c)`: returns Unicode (ASCII) of the character.
- Example: `ord('a')` returns 97.
- `chr(x)`: returns the character whose Unicode is `x`.
- Example: `chr(97)` returns ‘a’.
In Pairs or Triples...

Some review and some novel challenges:

```python
#Predict what will be printed:

for c in range(65,90):
    print(chr(c))

message = "I love Python"
newMessage = ""
for c in message:
    print(ord(c))  #Print the Unicode of each number
    print(chr(ord(c)+1))  #Print the next character
    newMessage = newMessage + chr(ord(c)+1)  #add to the new message
print("The coded message is", newMessage)

word = "zebra"
codedWord = ""
for ch in word:
    offset = ord(ch) - ord('a') + 1  #how many letters past 'a'
    wrap = offset % 26  #if larger than 26, wrap back to 0
    newChar = chr(ord('a') + wrap)  #compute the new letter
    print(wrap, chr(ord('a') + wrap))  #print the wrap & new letter
    codedWord = codedWord + newChar  #add the newChar to the coded w
print("The coded word (with wrap) is", codedWord)
```
Python Tutor

```python
# Predict what will be printed:
for c in range(65,90):
    print(chr(c))
message = "I love Python"
newMessage = ""
for c in message:
    print(ord(c))  # Print the Unicode of each number
    print(chr(ord(c)+1))  # Print the next character
    newMessage = newMessage + chr(ord(c)+1)  # Add to the new message
print("The coded message is", newMessage)
word = "zebra"
codedWord = ""
for ch in word:
    offset = ord(ch) - ord('a') + 1  # How many letters past 'a'
    wrap = offset % 26  # If larger than 26, wrap back to 0
    newChar = chr(ord('a') + wrap)  # Compute the new letter
    print(wrap, chr(ord('a') + wrap))  # Print the wrap & new lett
    codedWord = codedWord + newChar  # Add the newChar to the coded word
print("The coded word (with wrap) is", codedWord)
```
User Input

Covered in detail in Lab 2:

```python
1  mess = input('Please enter a message: ')
2  print("You entered", mess)
```

(Demo with pythonTutor)
Side Note: ‘+’ for numbers and strings

- \( x = 3 + 5 \) stores the number 8 in memory location \( x \).
Side Note: ’+’ for numbers and strings

- \( x = 3 + 5 \) stores the number 8 in memory location \( x \).
- \( x = x + 1 \) increases \( x \) by 1.
Side Note: ‘+’ for numbers and strings

- $x = 3 + 5$ stores the number 8 in memory location $x$.
- $x = x + 1$ increases $x$ by 1.
- $s = "hi" + "Mom"$ stores "hiMom" in memory locations $s$. 

Side Note: ‘+’ for numbers and strings

- $x = 3 + 5$ stores the number 8 in memory location $x$.
- $x = x + 1$ increases $x$ by 1.
- $s = "hi" + "Mom"$ stores ”hiMom” in memory locations $s$.
- $s = s + "A"$ adds the letter $x$ to the end of the strings $s$. 
1. (a) What will the following Python code print:

```python
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
half = months[6]
print(half.upper())
print(half[0])
print(months[-1].lower())
print(months[2:4])
start = 9
print(months[start-1])
term = 3
print(months[(start+term-1)%12])
```
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:
  - For-loops
  - `range()`
  - Variables: ints and strings
  - Some arithmetic
  - String concatenation
  - Functions: `ord()` and `chr()`
Recap

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

- In Python, we introduced:
  - For-loops
Recap

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

- In Python, we introduced:
  - For-loops
  - range()
Recap

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

- In Python, we introduced:
  - For-loops
  - `range()`
  - Variables: ints and strings
Recap

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

- In Python, we introduced:
  - For-loops
  - `range()`
  - Variables: ints and strings
  - Some arithmetic
On lecture slip, write down a topic you wish we had spent more time (and why).

In Python, we introduced:

- For-loops
- `range()`
- Variables: ints and strings
- Some arithmetic
- String concatenation
Recap

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

- In Python, we introduced:
  - For-loops
  - `range()`
  - Variables: ints and strings
  - Some arithmetic
  - String concatenation
  - Functions: `ord()` and `char()`
Lecture Slips & Writing Boards

- Turn in lecture slips & writing boards as you leave...