Questions/Announcements

- Python if (conditional/selection) statements
- Python data type: Boolean
- Python relational operators

How Much? Write a function, shipping, which determines the shipping cost of an order. Shipping is $.05 per pound but for shipments over 100 pounds, subtract $1.50

Python if statements

```
>>> a = 5
>>> b = 3

>>> if a > b:
    c = 10
    d = c * 2
>>> c
```
Python if statements

```python
>>> a = 5
>>> b = 3
>>> if a > b:
    c = 10
elif a < b:
    c = 15
else:
    c = 20
>>> c
```

Python built in data types:

- Integers
- Floating point
- String
- **Boolean**

**Truth Tables**

<table>
<thead>
<tr>
<th>not</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td></td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td></td>
<td>True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>and</th>
<th>or</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>xor</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>
Truth Tables

cold  windy  not cold and windy
T    T    T
T    F    F
F    T    T
F    F    F

Relational operators return Boolean values

<  
>  
<= 
>= 
== 
!=

>>> 'a' < 'b'
>>> 'apple' < 'banana'
>>> 'twenty-five' > 'twenty'

So do some string methods...

   endswith, startswith, islower, isupper
and a relational operator for sequences: in

```python
>>> 'a' in 'abc'
True
>>> 'bc' in 'abc'
True
>>> 'cde' in 'abc'
False
```

Be careful!

```python
>>> 'Apple' < 'banana'
>>> 'apple' < 'Banana'
>>> 'Zebra' < 'banana'
```

```python
>>> x = 100.0; y = 100; x == y
>>> x = 1/3; y = .33333333; x == y
```

Write a Python statement that sets a ‘flag’ to be True when CIS 122 lecture is happening.

```python
if day == 'Tuesday' or day == 'Thursday' and ('0830' <= time <= '0950'):
    CIS122 = True

print(CIS122)
```

day = 'Tuesday'; time = '1200'
day = 'Thursday'; time = '0900'

Order of Operations

- parentheses!
- arithmetic – **, unary +, - / // %, + -
- relational – all at same level of precedence – < > <= >= == != [not] in
- boolean – not and or

for same level of precedence – left to right
def check(n):
    '''what does this function do?'''
    if (n % 2) == 0:
        return True
    else:
        return False

>>> check(100)
>>> check(101)

def check(n):
    '''what does this function do?'''
    return (n % 2) == 0:

>>> check(100)
>>> check(101)

Payday: Write a function, payday, with parameters hourly_wage and hours. The function should compute and return the pay. For the first 40 hours worked, pay is at the input hourly wage. Any hours over forty but less than or equal to 60 are paid at 1 1/2 times the regular hourly wage. Any hours over sixty are paid at 2 times the regular hourly wage.