Python Cheat Sheet

Python Program Structure

- A Python *program* consists of a number of *statements* (just like an essay consists of a number of sentences)
- A *statement* consists of a number of *expressions*, together with *operators* and keywords (just like a sentence consists of a number of words, together with punctuations)

Running Python Programs

- When you *run/execute* a Python program, each *statement* is *interpret*ed. In order to interpret an statement, each of its expressions are to be *evaluat*ed.
- All expressions evaluate to some *value* of different *types*. Possible *types* of values are (with examples of values of that type):
  - *Numbers*: 1, 2, 3, -1, 4.5, -3.1, …
  - *Strings*: 'I like apples.', …
  - *Lists*: [1,2,3,], [1, 'abc'], …
  - *Booleans*: True, False
  - File objects: things returned by open(x)
  - Other objects that you do not (need to) know
- All of these values are themselves expressions (see next)

Expressions

- Basic Expressions:
  - *Numbers*: 1, 2, 3, -1, 4.5, -3.1, …
  - *Strings*: 'I like apples.', …
  - *Lists*: [1,2,3,], [1, 'abc'], …
  - *Booleans*: True, False
  - File objects: things returned by open(x)
  - Other objects that you do not (need to) know
- Compound Expressions:
  - 1+2, 3 > 5, ('ab' == 'ab') and (x != 4), …
- *Variables*:
  - Anything that consists of letters , digits and underscores (cannot start with a digit)
- *Function application*:
  - squre(3), max(x, 4), …

Statements

- *Assignment statements*
  - a = 5, b = a + a, a = a + 1
- *Function definition statements*

```
def cheerFor(team_name):
    print('Let's go ' + team_name + '!!!!!')
    return
```

- *Conditional statements*

```
if team == 'Brown Bears':
    cheerFor(team)
else:
    boo(team)
```

```
x = 4
y = 5

def max(a, b):
    if a > b:
        return a
    else:
        return b

m = max(12,45)
print(m)

m = max(x,y)
print(m)

m = max(x, 45)

print('the value of m is: ' + str(m))
```

assignment statements
(statements with the = sign)

function definition statements
(starts with def, then the body is like a mini program consisting of other statements)

function application. Notice how the values of the two arguments you provide get hooked to a and b within the function body.

Though function applications are expressions, they can appear here since they usually have 'side effects' when evaluated