Week 4

Intro Python

http://cadtx.pw/intropy4

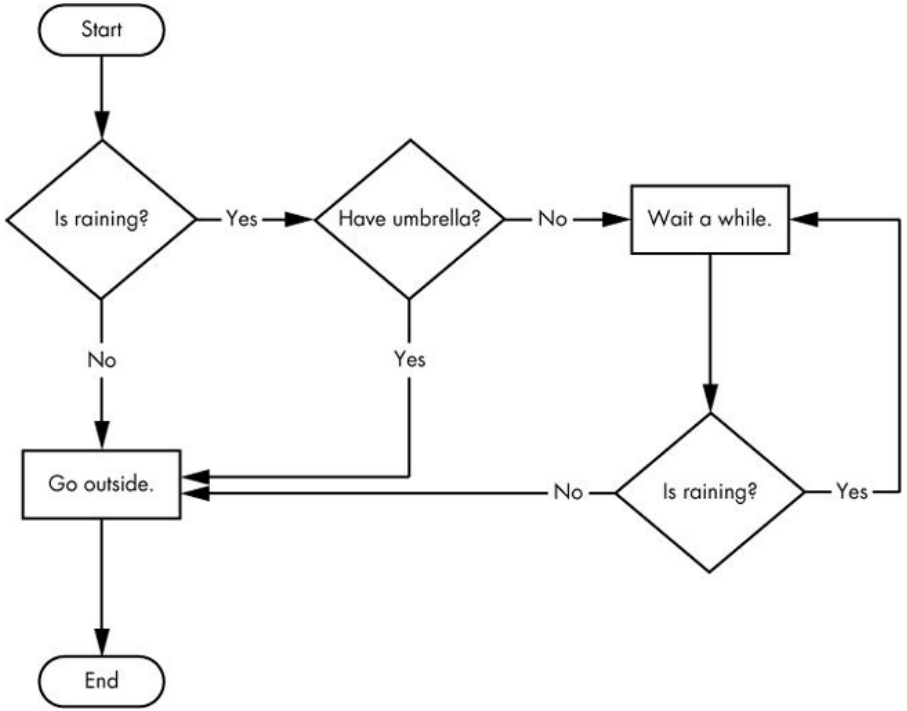
Objectives

Chapter 4

- Booleans
- Boolean Algebra
- If-elif-else statements

Control Structures and Flow

- So far our programs have been linear - one instruction after another
- This limits the strength of our programs
  - Using *flow control statements* we can alter how our programs execute



Booleans

- Before we can learn about flow control statements, we need to be able to represent these yes and no options
- We can do this using *Boolean* values
- While ints, floats, and strings have an unlimited number of possible values, booleans only have two:
  - *True*
  - *False*
- You define them in Python as *True* or *False* with no quotes.

```
1   # Defining booleans in python
2   var1 = True
3   var2 = False
4   var3 = "True"
5   print(var1, type(var1))
6   print(var2, type(var2))
7   print(var3, type(var3))
```

```
(True, <type 'bool'>)
(False, <type 'bool'>)
('True', <type 'str'>)
```

Comparison Operators

- Comparison operators take two values and return a single Boolean value

==	Equal to (**careful not to use '='!)**
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

```
In [1]: 42 == 42
Out[1]: True

In [2]: 42 == '42'
Out[2]: False

In [3]: 42 = 42.0
  File "<ipython-input-3-87d88c797f68>", line 1
    42 = 42.0
       ^
SyntaxError: can't assign to literal


In [4]: 42 == int("42")
Out[4]: True

In [5]: 'dog' != 'cat'
Out[5]: True

In [6]: 'Hello' == 'hello'
Out[6]: False

In [7]: 34 > 30
Out[7]: True

In [8]: 34 > 34
Out[8]: False

In [9]: 34 >= 34
Out[9]: True
```

Boolean operators

- There are three Boolean operators. They take boolean values and return a boolean. And and or take two values or expressions and are considered *binary*
  - and
    - evaluates True if and only if both values are True, otherwise evaluates to False.
  - or
    - Evaluates to True if *either* of the two values are True. If both are False, evaluates to False.
  - Not
    - Only takes on boolean value or expression
    - Returns the opposite Boolean value

```
In [10]: True and True
Out[10]: True

In [11]: True and False
Out[11]: False

In [12]: False and False
Out[12]: False

In [13]: True or True
Out[13]: True

In [14]: True or False
Out[14]: True

In [15]: False or False
Out[15]: False

In [18]: not True
Out[18]: False
In [19]: not False
Out[19]: True
```

Mixing Boolean and Comparison Operators

- Since comparison operators evaluate to Boolean values, you can use them with Boolean operators
- and, or, not all operate on the Boolean values *True* and *False*
- While 4 < 5 isn't a Boolean value, it evaluates down to one and can be used with Boolean operatrs

```
In [20]: (4 < 5) and (5 < 6)
Out[20]: True

In [21]: (4 < 5) and (9 < 6)
Out[21]: False

In [22]: (1 == 2) or (2 == 2)
Out[22]: True
```

Python will evaluate the left expression, then the right. When it knows the Boolean value for each it will then evaluate the whole expression down to a single Boolean value,

Comparing variables and checking lists

- The primary use of boolean and comparison operators is to compare the value of a variable to a certain value of interest
- You can also check if values are (or aren't!) in lists

```
In [23]: age = 19

In [24]: age < 21
Out[24]: True
```

```
In [26]: planets
Out[26]: ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

In [27]: 'Pluto' in planets
Out[27]: False

In [28]: :(

In [28]: 'Earth' in planets
Out[28]: True
```

If statements

- The simplest if statement has one conditional test and one action
- The formatting is similar to *for* loops
  - Whatever is indented will be executed provided the test evaluates to *True*

```
1  if conditional_test:
2      do something
```

```
1  age = 19
2  if age >= 18:
3      print("You are old enough to vote")
```

```
Running: W4.py
You are old enough to vote
>>>
```

if-else statements

- Oftentimes you'll want to take one action when a test passes and a different one in all other cases.
- If the case on line 2 passes, it will execute the first block of code and end the program.
- If it fails and evaluates to *False* the the else block on line 5 is executed.

```
1  age = 17
2  if age >= 18:
3      print("You are old enough to vote")
4      print("Have you registered to vote yet?")
5  else:
6      print("Sorry, you are too young to vote.")
```

Running: W4.py

```
Sorry, you are too young to vote.
>>>
```

```
1  age = 12
2  admission_cost = 0
3  if age < 4:
4      admission_cost = 0
5  elif age < 18:
6      admission_cost = 5
7  else:
8      admission_cost = 10
9  print("Your admission cost is ${}".format(admission_cost))
```

```
Running: W4.py
Your admission cost is $5
>>>
```

- Oftentimes you'll want to test more than two possible situations.
- Python will only execute one block in an if-elif-else chain.
- Consider a theme park with the following admission rates
  - Under age 4 is free entry.
  - Anyone between 4 and 18 is $5
  - Anyone 18 or older is $10

TRY IT YOURSELF

5-3. Alien Colors #1: Imagine an alien was just shot down in a game. Create a variable called `alien_color` and assign it a value of `'green'`, `'yellow'`, or `'red'`.

- Write an `if` statement to test whether the alien's color is green. If it is, print a message that the player just earned 5 points.
- Write one version of this program that passes the `if` test and another that fails. (The version that fails will have no output.)

5-4. Alien Colors #2: Choose a color for an alien as you did in Exercise 5-3, and write an `if-else` chain.

- If the alien's color is green, print a statement that the player just earned 5 points for shooting the alien.
- If the alien's color isn't green, print a statement that the player just earned 10 points.
- Write one version of this program that runs the `if` block and another that runs the `else` block.

5-5. Alien Colors #3: Turn your `if-else` chain from Exercise 5-4 into an `if-elif-else` chain.

- If the alien is green, print a message that the player earned 5 points.
- If the alien is yellow, print a message that the player earned 10 points.
- If the alien is red, print a message that the player earned 15 points.
- Write three versions of this program, making sure each message is printed for the appropriate color alien.

5-6. Stages of Life: Write an `if-elif-else` chain that determines a person's stage of life. Set a value for the variable age, and then:

- If the person is less than 2 years old, print a message that the person is a baby.
- If the person is at least 2 years old but less than 4, print a message that the person is a toddler.
- If the person is at least 4 years old but less than 13, print a message that the person is a kid.
- If the person is at least 13 years old but less than 20, print a message that the person is a teenager.
- If the person is at least 20 years old but less than 65, print a message that the person is an adult.
- If the person is age 65 or older, print a message that the person is an elder.

http://cadtx.pw/week4kahoot