Variables

Clicker #0

Please answer this iClicker question.

- a) okay, sure
- b) yes, of course!
- c) I don't want to
- d) why are we doing this

Programming With Python

Programming: a way to ask computer to store values (variables), and do things with them (operations).

In [1]:
```
# This is a comment. You can write a comment by using a `#`

my_variable = 12

my_other_variable = 13 # Comments can be 'inline', like this one
```

Defining Variables

In programming, variables are things that store values. Variables are defined with `name = value`.

In [2]:
```
my_var = 1  # `my_var` is a variable

# This defines another variable
other_var = 'variables are cool'
```

Code Variables != Math Variables

In mathematics: = refers to equality (as a statement of truth).

In coding: = refers to assignment.

Math: What is x?

```
y = 10x + 2
```

Code: What is x?

```
x = x + 1
```

Clicker #1

After executing the following code, what will be the value of `my_var` ?

```
my_var = 2

my_var = my_var + 1
```

- a) 2
- b) 3
- c) "my_var + 1"
- d) This code will fail

Clicker Question Answer

In [3]:
```
my_var = 2
my_var = my_var + 1
print(my_var)
```

```
3
```

- In programming = means assignment
- Anything to the right of the = is evaluated before assignment

Namespace

The namespace is the 'place' where all your currently defined code is declared - all the things you have stored in active memory.

In [4]:
```
# You can list everything declated in the namespace with '%whos'
%whos
```

```
Variable            Type    Data/Info
-----------------------------------
my_other_variable   int     13
my_var              int     3
my_variable         int     12
other_var           str     variables are cool
```

Declaring Variables Cheat Sheet

- Names are always on the left of the =, values are always on the right
- Names are case sensitive
- Variables must start with letters
    - After that, they can include numbers, and underscores
    - They cannot include special characters (like &, *, #, etc)
- Python doesn't care what you name you variables
    - Humans do care. Pick names that describe the data / value that they store

Clicker #2

After executing the following code, what will be the value of `var_2` ?

```
var_2 = var_1 = 1
```

- a) 'var_1'
- b) 1
- c) 2
- d) This code will fail

Clicker Question Answer

In [5]:
```
var_2 = var_1 = 1
print(var_2)
```

```
1
```

- There can be more than one assignment in a single line
- Anything to the right of the = is evaluated before assignment
    - This process proceeds from right to left

Variable Types

Every variable has a 'type', which refers to the kind of variable that it is, and how the computer stores that data.

In [6]:
```
# Declare a variable
variable_name = 1

# You can always ask Python 'what type is this variable' using:
type(variable_name)
```

Out[6]:
```
int
```

Int

`Integers` store unsigned, whole numbers.

In [7]:
```
my_integer = 1
another_integer = 321
```

Float

`Floats` store signed, decimal-point numbers.

In [8]:
```
my_float = 1.0
another_float = -231.45
```

String

`Strings` store characters, as text.

In [9]:
```
my_string = 'words, words, words'
another_string = 'more words.'
```

Boolean

`Booleans` store `True` or `False`.

In [10]:
```
my_bool = True
another_bool = False
```

None

`None` is a special type that stores `None`, used to denote a null or empty value.

In [11]:
```
the_concept_of_nothing = None
```

Clicker #3

After executing the following code, what will the value of `var_b` be?

```
var_a = 1

var_b = var_a
```

- a) var_a
- b) 'b'
- c) 1
- d) This code will fail

Clicker Question Answer

In [12]:
```
var_a = 1
var_b = var_a

print(var_b)
```

```
1
```

- Multiple variables can relate to the same value(s)

Aliases

Variables are names assigned to a value. Values can have more than one name.

In [13]:
```
# Make a variable, and an alias
a = 1
b = a

print(b)
```

```
1
```