Programming in Python

Variables

Variables

- You have probably seen variables in a math class.

```
x = 3 + 5
x = 8
```

```
y = 7
z = y * 4
z = 28
```

- Variables in programming are similar, but we have different types (not just numeric).

Variables

- We use variables in computer programs to allow us to store information.
- Each variable allows us to access memory in the computer where we can put data.
- Each variable will have a name (or label).
- The information that the variable accesses can change or vary, hence the word "variable".

Variables

- Syntax (pseudocode)

```
variable = value
```

```
variable = expression
```

- Code

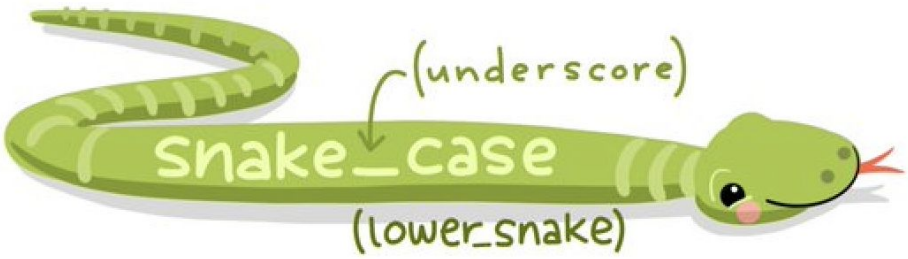
```
age = 20
```

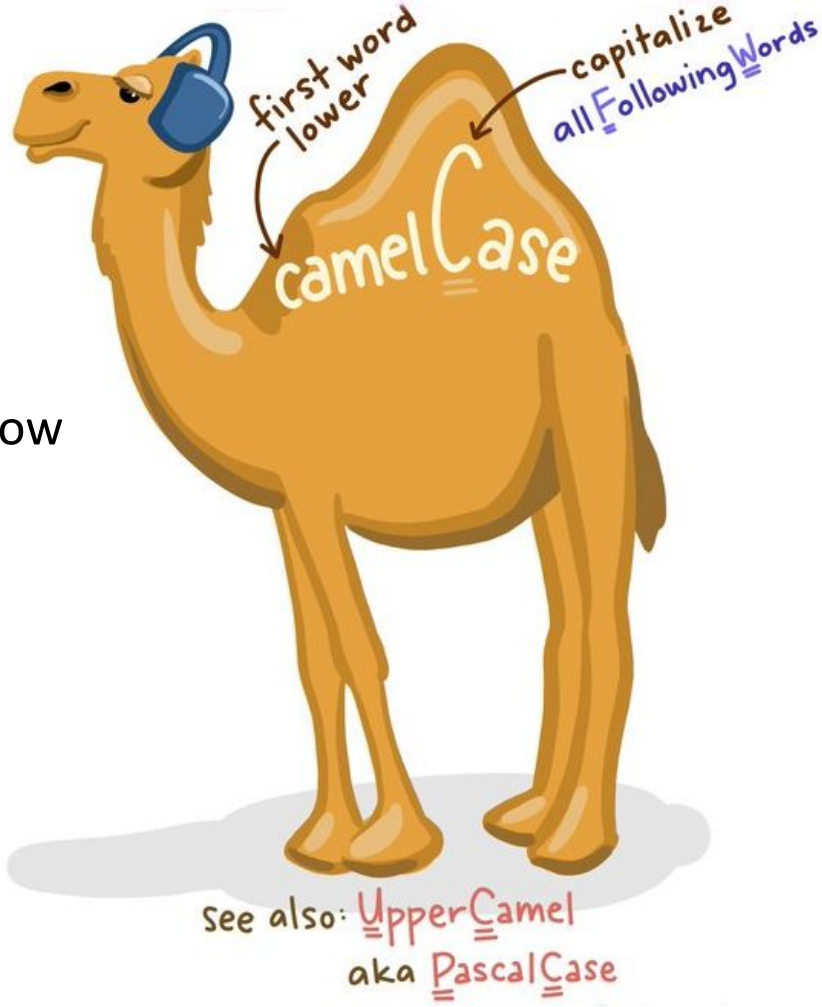
```
total = 5 + 3
```

- The = (equals sign) means assignment.
  - › Take the number **20** and store it in a variable named **age**.
- The variable has to be on the left side of the =.
  - › The computer will evaluate the expression on the right side.
  - › Then the variable will be able to access that information.

Valid Variable Names

- Start with a letter (a-z, A-Z) or underscore (_)
- Can contain digits (0-9)
- No spaces or other characters
- Two conventions:
  - camelCase
    - Used in lots of languages
    - Examples: numStudents, isRainingNow
  - snake_case
    - More Pythonic
    - Examples: num_students, is_raining





Primitive Variable Types

- Python has various types of variables
  - Primitive types (we will learn today)
  - Non-primitive types (we will learn in a future level)
- The primitive data types in Python are:

Data Type	Description	Examples
`str`	a string of characters	`greet = "Hello World!"`

`int`	an integer or whole number that can be positive or negative	`num = 42`
		`zero = 0`
		`negative = -1`

`float`	a floating point or decimal number	`pi = 3.1415` `neg = -12.5`
`bool`	a boolean has a value of True or False	`isLearning = True` `isSnowing = False`

Variables

- Variables have 3 parts
  - name (or label)
  - type (such as str, int, float, bool)
  - value (determines the type)
- Create a variable

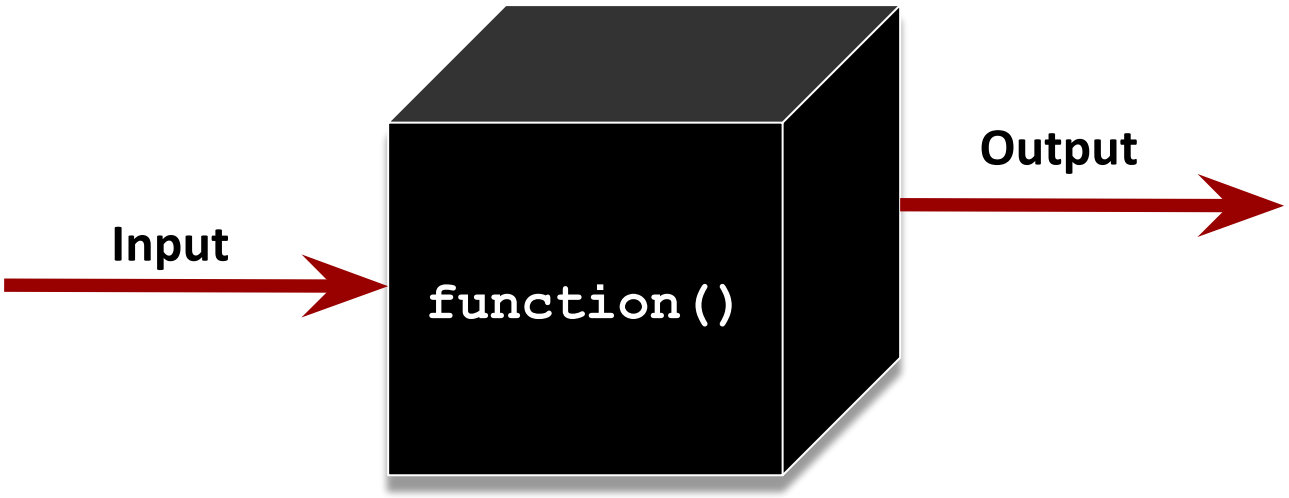
```
variable = value
```

- Examples

```
greeting = "Good day"
course = "2: Python"
year = 2023
half = 0.5
isCool = True
```

Functions

- Perform a task and then return control to your program
- Reuse code
  - Write once, use many times
- Many functions have been defined that we can use



Print function

```
print(object(s), sep=separator, end=end)
```

Parameter	Description
object(s)	Any object, and as many as you like. Will be converted to string before printed
sep	Optional. Specify how to separate the objects, if there is more than one. Default is ' ' (one space)
end	Optional. Specify what to print at the end. Default is '\n' (new line)

- When we call the print() function, anything we put in the parenthesis () is input.
- There is no output.

Print Using Variables

```
school = "USC"
age = 140
print(school, age)
print(school, "is", age, "years old")
```

```
USC 140
USC is 140 years old
```

```
author = "Mahatma Gandhi"
quote = "My life is my message"
print(author, " said, \"", quote, ".\"", sep="")
```

```
Mahatma Gandhi said, "My life is my message."
```

Print Examples

```
num1 = 20
num2 = 23
print(num1, num2, sep="")
```

```
2023
```

```
msg = "Happy New Year"
year = 2023
print(msg, year, sep="*")
```

```
Happy New Year*2023
```

```
print("Hello", end=" ")
print("World")
print("Python is fun!")
```

```
Hello World
Python is fun!
```

What is a comma for?

- Using commas is NOT string concatenation.
- You can use commas when calling the print() function (or any function) to give it multiple pieces of information (i.e., arguments).
- For print(), the arguments are the things to print such as strings and integers.

Simple Math Example

- Let's do an example of using a variable.
- Write a program that solves the equation:
  x = 3 * 4 + 7

```
x = 3 * 4 + 7
print(x)
```

```
result = 3 * 4 + 7
print(result)
```

- If you give the print() function a variable, it will print the value of the variable!

```
19
```

Arithmetic Operators

- To store numbers, Python has two types of variables: **`int`** and **`float`**.
- We can do mathematical operations with each.
- Arithmetic operators that are the similar to a calculator:

+	−	×	÷

+	-	*	/

- Modulo (remainder from division):

%

- Exponent:

**

String Concatenation

- Use the plus (+) to add two strings together is string concatenation.
- We are concatenating two strings together.
- The result is a string (str).

```
code = "TECH"
num = "2"
print(code + num)
course = code + "-" + num
print(course)
```

```
TECH2
TECH-2
```

String Concatenation

- You cannot combine a number and a string together.
- The Python interpreter (built into PyCharm) will NOT allow it.
- If you want to concatenate them together, then you need to convert the number to a string.

```
code = "TECH"
num = 2
print(code + str(num))
course = code + "-" + str(num)
print(course)
```

```
TECH2
TECH-2
```

Convert to String

- Use can use the str() function on lots of types of variables including
  - int
  - float
  - bool

```
weather = "It is snowing"
isSnowing = False
print("\"" + weather + "\" is " + str(isSnowing))
```

```
"It is snowing" is False
```