

## 1 Goals

- Basic introduction to Computer Science
- Basic introduction to Python
- Reading documentation and using Google
- Simple data analysis in Python

## 2 What is Computer Science?

- When I say Computer Science, what do you think of?
- Is it a discipline or a tool?
- Do you need a computer to study it?

### 2.1 Formal Definition

Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures that underlie the acquisition, representation, processing, storage, communication of, and access to, information.

### 2.2 So...

- Is it a discipline or a tool?
- Do you need a computer to study it?
- Disclaimer

## 3 Let's Begin

1. Open IDLE and create a new file
2. Save it as something memorable
3. Type:

```
print('Hello World!')
```

4. Run it
5. Be greeted!

### 3.1 But wait, what'd that do

- From before:

```
print('Hello World!')
```

- `print()` is a function
- `'Hello World!'` is an argument
- So, to generalize:
  - `print(arg)` prints the `arg`

## 4 Variables

- Does `print(Hello world!)` work?
- Predict first
- Test it
- Analyze the error
  - What's the error?

### 4.1 Strings

- A string is any expression of characters enclosed in `'` or `"`
- Is `'h'` a string?
- How about `h`?
- What about `'3'`?
- How about `3`?
- How about `'Hi, what's up Bob?'`?
- What about `"Hi, what's up Bob?"`?
- So what variable type does the `print` function accept?

## 4.2 Numbers

- A number is a representation of, well, a number...
- 3 is a number, '3' is not
- You can do math on numbers!
- 3 + 3 is valid Python
- So is 3 - 3, 3 \* 3, 3 / 3, and even 3 \*\* 3.
- What is \*\*
- Technically speaking, these are function shortcuts
  - 3 + 3 is really `add(3, 3)`
- Try some stuff out
- Does Python understand parentheses?

## 4.3 Ok, but what is a variable?

It's kinda like a big box

- Let's see it in action:  

```
x = 1
y = 2

z = x + y
print(z)

print(x + y)
```
- Let's go through our process again:
  - Predict
  - Test
  - Analyze

## 4.4 Same drill

Predict, test, analyze...

- Here's another one for you

```
x = 'this'
y = 'is'
z = 'a word'
```

```
nstring = x + y + z
print(nstring)
```

- Given above:

```
print(x + y + z)
```

- And finally:

```
print(x + ' ' + y + ' ' + z)
```

## 5 Boolean Expressions

Just another variable type...

- Can only be **True** or **False**
- What operators could we use to get boolean values?
- How about to combine two boolean values to one?
- Let's check this out:

```
a, b = 1, 2
```

```
x, y = True, False
```

```
print(a < b)
```

- How about:

```
print(x and y)
print(x or y)
print (x and not y)
```

## 5.1 Boolean Operators

Non-Boolean Variables  $\rightarrow$  Boolean Variables:

	Operator	Description
	<	Less than
	>	Greater than
•	==	Equality
	<=	lte
	>=	gte
	!=	Not equals

## 5.2 Boolean Operators

Boolean Variables  $\rightarrow$  Boolean Variables:

	Operator	Description
•	and	logical and
	or	logical or
	not	logical not

## 5.3 Um, why can't we just use = for equality

Like normal people...

- Do we already have = in python?
- What does it do?
- Consider:

```
x = 1
```

```
x == 1
```

- Do we want them to be interchangeable?

## 6 Control Structures

If this, then that

- This is like a fork in the road

- What's a good way to represent whether or not to execute code? Variable type?
- Consider

```
x, y = 1, 2

if (x < y):
    print('x is less than y')
```

- When will it execute the print statement?

## 6.1 If, elif, and else

- What's the difference here, is there one?

```
x, y = 1, 2

if (x < y):
    print ('x is less than y')

if (x < y):
    print ('Yipee')

if (x > y):
    print('Nope')

x, y = 1, 2

if (x < y):
    print('x is less than y')

elif (x < y):
    print('Yipee')

else:
    print('Nope')
```

## 7 While loops

Same deal really...

- Instead of executing the block if true once, it repeats until the condition is `False`
- What is printed out?

```
i = 0
```

```
while (i < 100):
    print (i)
    i = i + 1
```

- Let's trace through it

## 8 How to Google

What if I forget something, is it ok to Google it?

- Let's try and figure out what a function `input()` does
- First step, try the official python documentation:

**`input([prompt])`**

If the *prompt* argument is present, it is written to standard output without a trailing newline. The function then reads a line from input, converts it to a string (stripping a trailing newline), and returns that. When EOF is read, `EOFError` is raised. Example:

```
>>> s = input('--> ')
--> Monty Python's Flying Circus
>>> s
'Monty Python's Flying Circus'
```

If the `readline` module was loaded, then `input()` will use it to provide elaborate line editing and history features.

- So, let's analyze this, and try it

### 8.1 How to Google well

- Is it better to Google smaller questions or the entire problem?
  - Why?
- Where's the line between plagiarizing and utilizing the community?
- What are some good websites for looking stuff up on?
- Try:
  - Stack overflow
  - Official Python Documentation

## 9 Project 1

A calculator

1. Ask the user to select an operation
2. Ask the user for the inputs of that operation
3. Perform the operation, print the result
4. Ask if the user wishes to use the calculator again
  - (a) If yes, repeat
  - (b) If no, exit

## 10 Project 2

A game

1. Computer picks a random number between 1 and 100
2. Computer asks user for a guess
3. Computer tells user whether they are high, low, or correct
4. Give the user 10 guesses
5. When the users guesses it, tell them how many guesses it took
6. If the user doesn't get it in 10 guesses, print what the number was

### 10.1 Project 2 Extension

- Can we teach the computer to intelligently make the best guess?
- Switch the roles, have you pick the number, and the computer guess?
- What's the maximum guesses it will take for the computer to get it in theory?