•••

Unit 1: Fundamentals of coding with python

Requirements

- Python installed
- Internet
- Passion to learn

BLACK DISRUPTOR ACCELERATOR

Session 1

What is Python?

- Python is an interpreted, high-level and general-purpose programming language.

- Python is used for:
  - Web development
  - Software Development
  - Mathematics
  - Scripting

What is an interpreted language?

- An interpreter takes the code you write and executes (runs) whatever actions you specified, creates the variables you created, and does a lot of behind-the-scenes work to ensure it runs smoothly or tells you about errors.

- Python is called an interpreted language because it goes through an interpreter, which turns code you write into the language understood by your computer's processor.

What is Python used for?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Python Syntax

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

How are you feeling?

RED

I have no idea what you're talking about

YELLOW

I have some questions but feel like I understand some things

GREEN

I feel comfortable with everything you've said

BLACK DISRUPTOR ACCELERATOR

IDE’s

Integrated Development Environment

- You can run Python code using an Integrated Development Environment (IDE).
- IDEs tend to contain many useful tools to make your development easier, if you know how to use them well! Examples of IDEs include VSCode and PyCharm.

Google Colab

Google Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

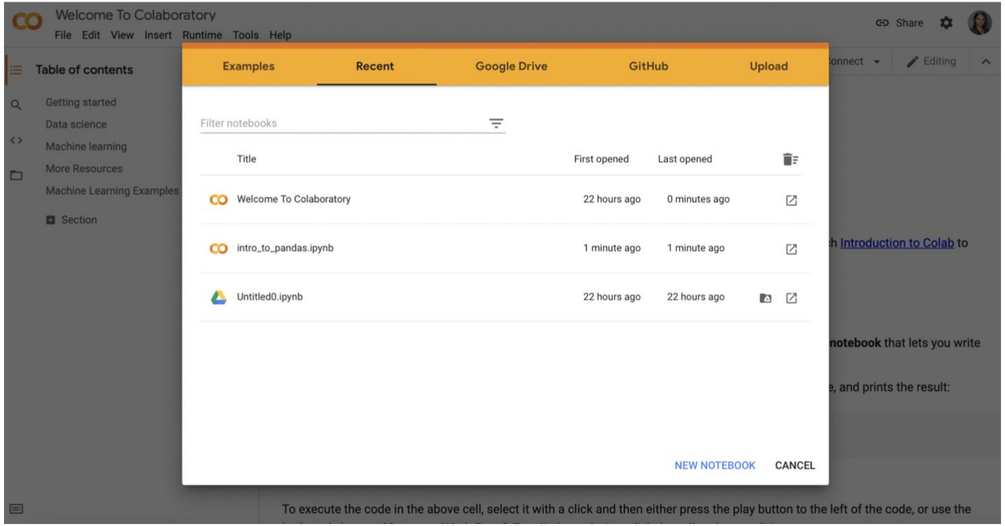
Can be used by anyone! This includes Beginners, Professional, Students, Data Scientists and researchers etc.

A very good alternative to Jupyter notebook, since you do not need to download it.

A notebook document is composed of cells, each of which can contain code, text, images, and more.

Google Colab

- Visit: https://colab.research.google.com
- Create a notebook for todays session (PythonBootcampDay1.ipynb) on Google Colab



Title	First opened	Last opened
Welcome To Colaboratory	22 hours ago	0 minutes ago
intro_to_pandas.ipynb	1 minute ago	1 minute ago
Untitled0.ipynb	22 hours ago	22 hours ago

h Introduction to Colab to

notebook that lets you write

, and prints the result:

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the

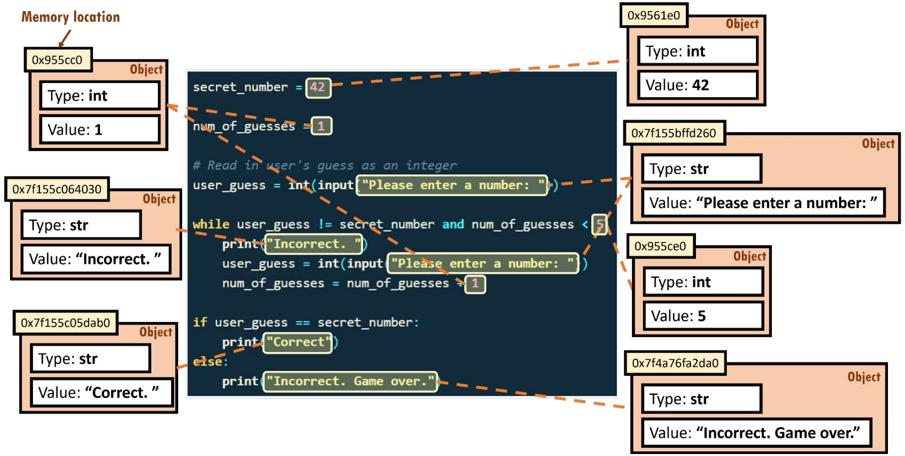
TASK 1: Google Collab

- Give the notebook a title and mini description (use the text section for this)
- Extension: Use python’s print function to write out your name and programming interests

Topic 1: Objects

In Python, every entity is an object.

- Each object is allocated some space in memory, and each object has a type and some value.
- Each object has a **type**



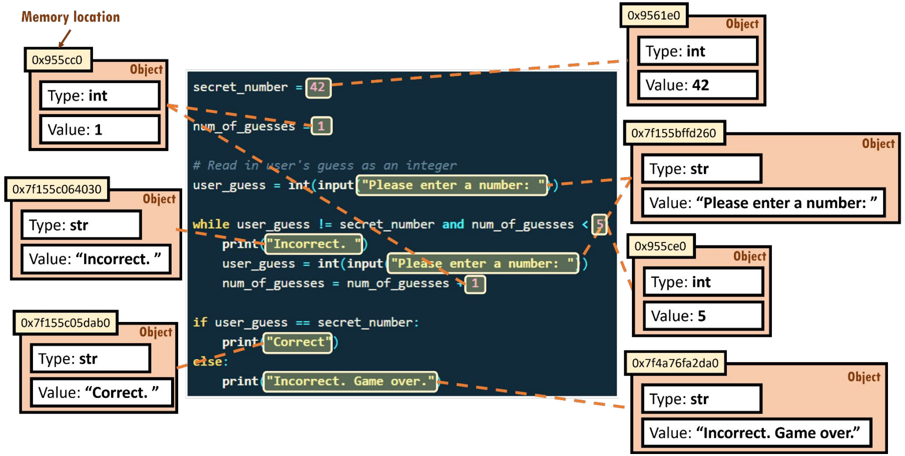
```
secret_number = 42
num_of_guesses = 1

# Read in user's guess as an integer
user_guess = int(input("Please enter a number: "))

while user_guess != secret_number and num_of_guesses < 5:
    print("Incorrect. ")
    user_guess = int(input("Please enter a number: "))
    num_of_guesses = num_of_guesses + 1

if user_guess == secret_number:
    print("Correct")
else:
    print("Incorrect. Game over.")
```

Python: Types

Python provides the following basic data types.

- Numbers: `int`, `float`, `complex`
- Boolean: `bool` (Something that is either `True` or `False`)
- Strings: `str` (A sequence of characters)

You can check the type of an object using `type()`.



```python
secret_number = 42

num_of_guesses = 1

# Read in user's guess as an integer
user_guess = int(input("Please enter a number: "))

while user_guess != secret_number and num_of_guesses < 5:
    print("Incorrect. ")
    user_guess = int(input("Please enter a number: "))
    num_of_guesses = num_of_guesses + 1

if user_guess == secret_number:
    print("Correct")
else:
    print("Incorrect. Game over.")
```

Types: Exercise

- **Type in the following code into a cell and then run:**

```
type(“London”)
type(3)
type(True)
```

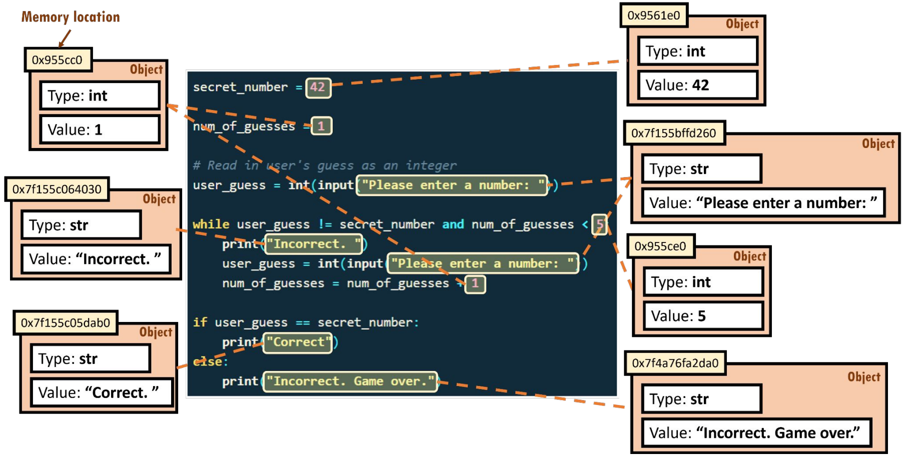
- **What do you see?**

Python: Types Q&A

What are the types of the following:

a. 3.142
b. 10000000
c. 1+2j
d. -3
e. True
f. False
g. true
h. 0
i. “Social Distancing”

You can check the type of an object using `type()`.



```python
secret_number = 42

num_of_guesses = 1

# Read in user's guess as an integer
user_guess = int(input("Please enter a number: "))

while user_guess != secret_number and num_of_guesses < 5:
    print("Incorrect. ")
    user_guess = int(input("Please enter a number: "))
    num_of_guesses = num_of_guesses + 1

if user_guess == secret_number:
    print("Correct")
else:
    print("Incorrect. Game over.")
```

Type: int
Value: 1

Type: str
Value: “Incorrect. ”

Type: str
Value: “Correct. ”

Type: int
Value: 42

Type: str
Value: “Please enter a number: ”

Type: int
Value: 5

Type: str
Value: “Incorrect. Game over.”

Demo

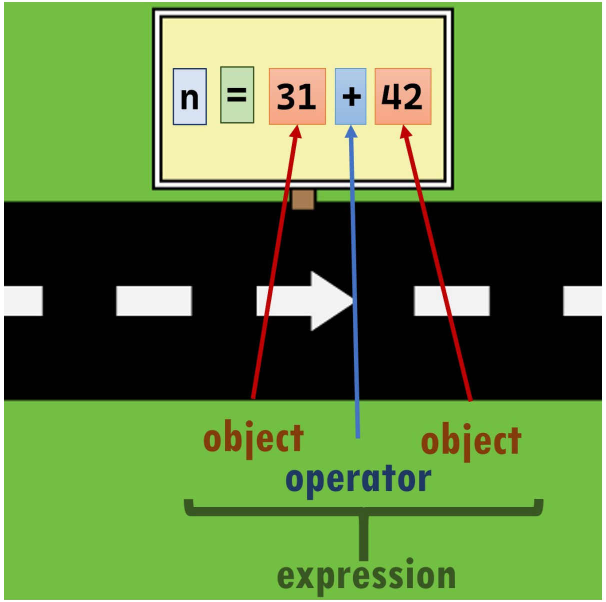
BLACK DISRUPTOR ACCELERATOR

BLACK DISRUPTOR ACCELERATOR

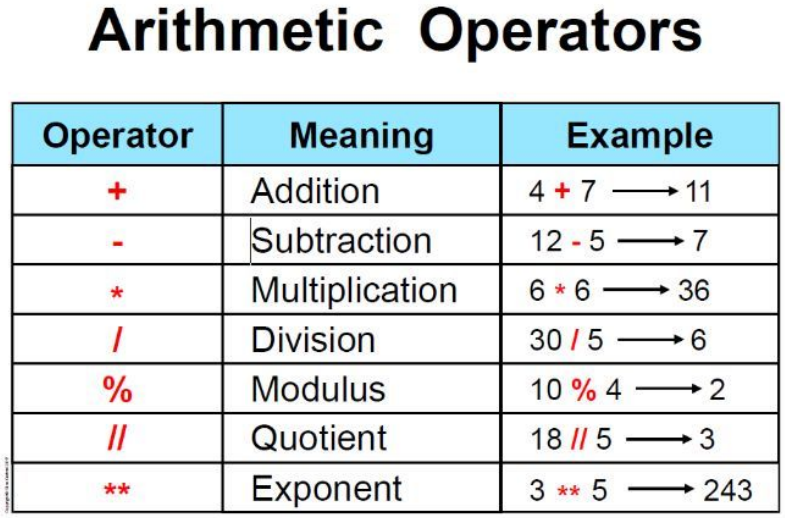
Topic 2: Operators and Expressions

Expressions

- You can combine objects with operators like **+** and **–** to form expressions.
- Expressions are something that can produce a value, for example **5+2** evaluates to **7**.

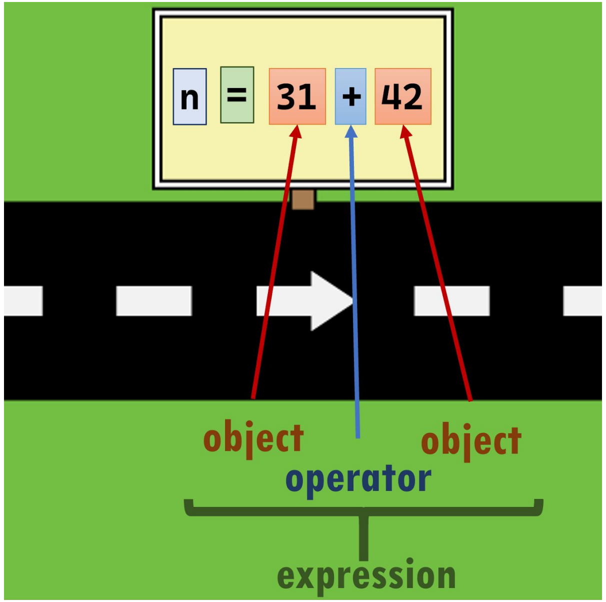


Python Arithmetic Operators



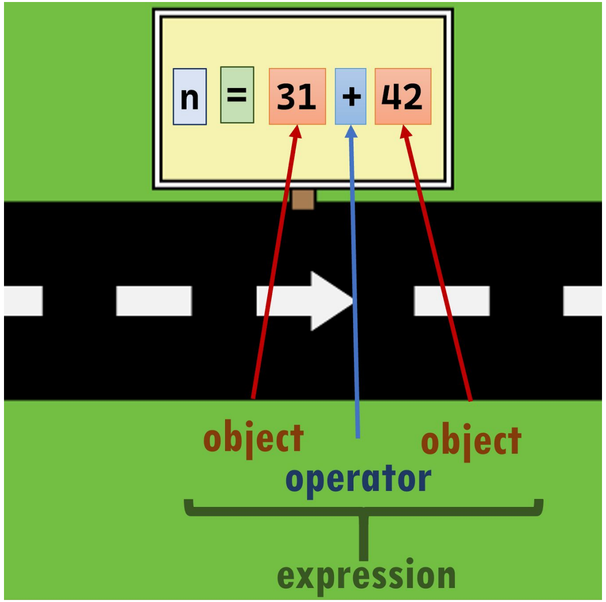
Arithmetic Operators

Operator	Meaning	Example
+	Addition	4 + 7 ⟶ 11
-	Subtraction	12 - 5 ⟶ 7
*	Multiplication	6 * 6 ⟶ 36
/	Division	30 / 5 ⟶ 6
%	Modulus	10 % 4 ⟶ 2
//	Quotient	18 // 5 ⟶ 3
**	Exponent	3 ** 5 ⟶ 243



Operator Precedence

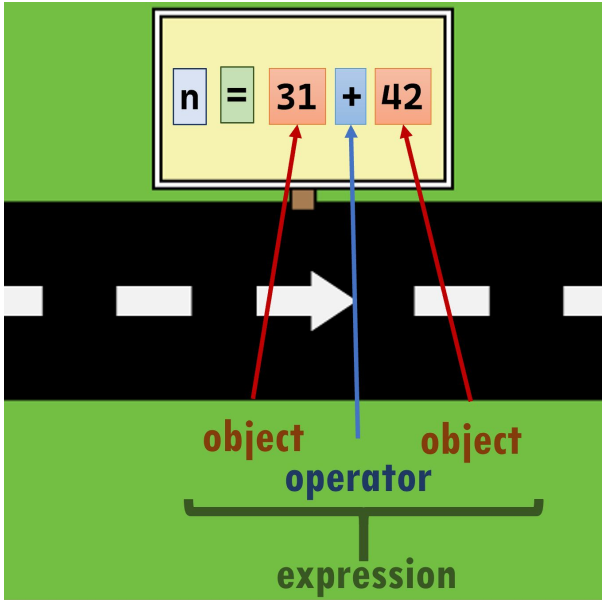
- You can chain expressions ie. 9 + 3 + 2
- The order of Operators mostly follow BIDMAS, as in mathematics.



QUESTIONS

- Evaluate the following the expressions in a cell in Google Colab:
  - 7 * 3 / 2
  - 7.3 / 2
  - 7 // 2
  - 7 % 2
  - 7 ** 2
  - 7 + 2 / 3 ** 2 - 5
  - “python” + “bootcamp”

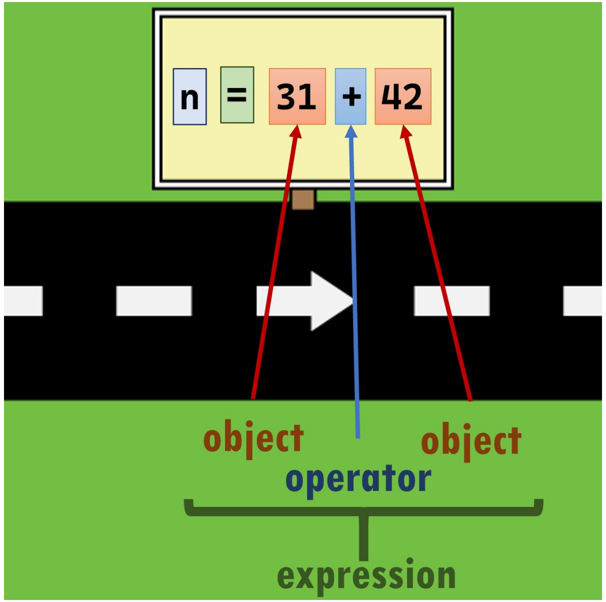
- **What answers do you get?**



Operators with Strings

- Some operators are not limited to arithmetic operations with numbers.
- For example, **+** and ***** can be used for a different kind of operation when used with **`strings`.**

- **Let us investigate.**

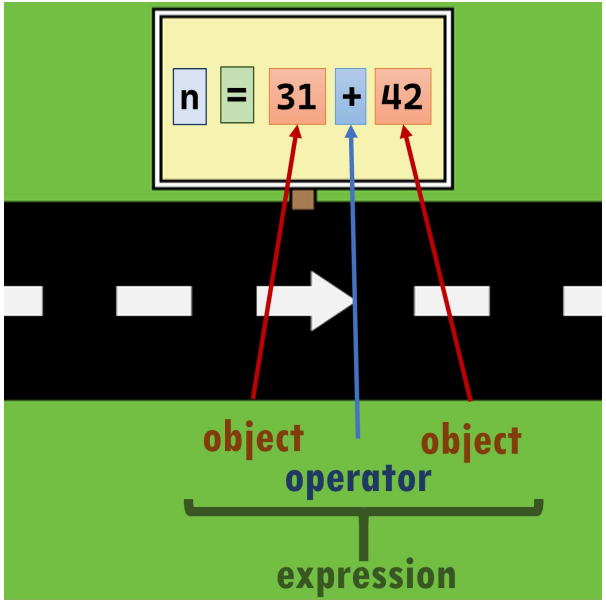


QUESTIONS: Part 2

- Evaluate the following the expressions in a cell in Google Colab:

    - “python” + 3
    - “python” * 3
    - 3 * “python”
    - “I love” + “ ” + “python” * 3
    - “I love” * 3 + “python”
    - (“I love python”) * 3

- **What answers do you get?**



Demo

How are you feeling?

RED

I have no idea what you're talking about

YELLOW

I have some questions but feel like I understand some things

GREEN

I feel comfortable with everything you've said

BLACK DISRUPTOR ACCELERATOR

Topic 3: Variables

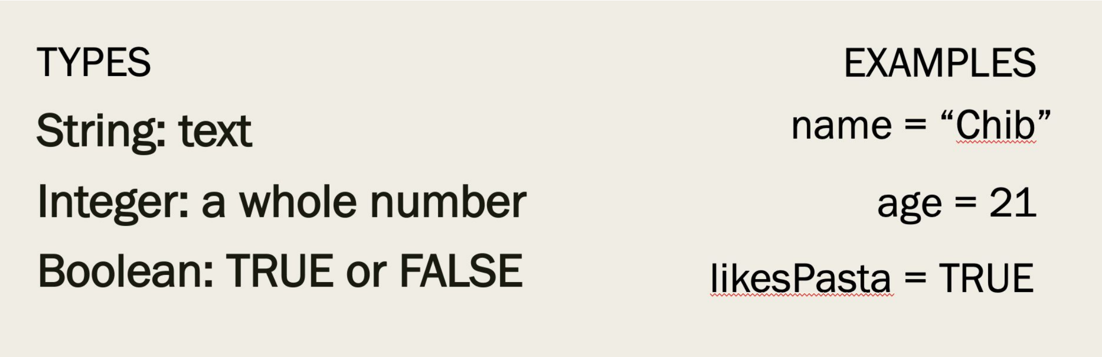
Variables

- Variable names can contain letters, numbers, and the underscore.
- Variable names *cannot* contain spaces.
- Variable names *cannot* start with a number.
- Case matters—for instance, temp and Temp are different.

It helps make your program more understandable if you choose names that are descriptive, but not so long that they clutter up your program.

Variables

- **A Python variable is a name that refers to an object.**
- **One of the major purposes of a variable is to remember a value from one part of a program so that it can be used in another part of the program.**

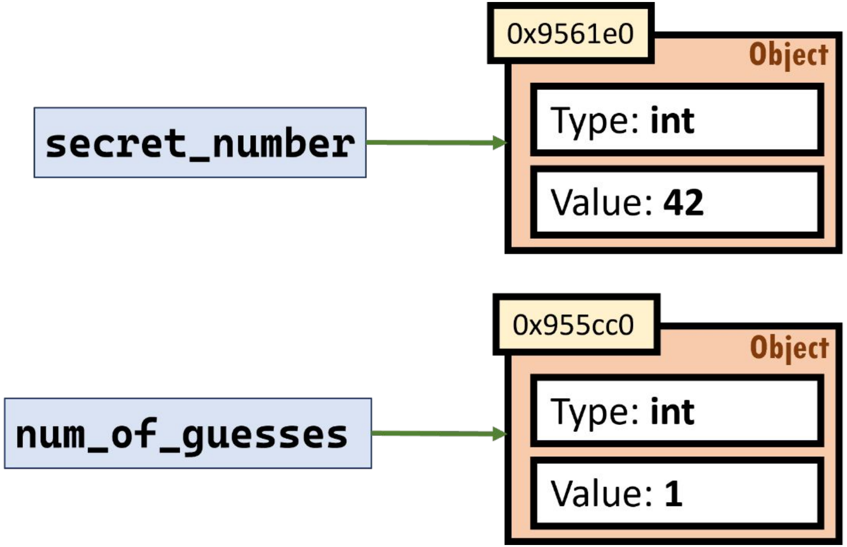


TYPES	EXAMPLES
String: text	name = “Chib”
Integer: a whole number	age = 21
Boolean: TRUE or FALSE	likesPasta = TRUE

Variables

- As we can see below, you can assign an object to a variable with an assignment statement.
- The assignment operator is ‘=’

```
secret_number = 42

num_of_guesses = 1
```



Rules for variable names

- Variable names can contain letters, numbers, and the underscore.
- Variable names *cannot* contain spaces.
- Variable names *cannot* start with a number.
- Case matters—for instance, temp and Temp are different.

It helps make your program more understandable if you choose names that are descriptive, but not so long that they clutter up your program.

Rules for variable names

- There are a list of reserved words in python that cannot be used as variable names.
- Here are some examples:
  a. False
  b. True
  c. None
  d. and
  e. as
  f. assert
  g. async
  h. await
  i. break
  j. class
  k. continue
  l. def
- Can you guess why?
- No need to memorise them! Python will complain if you try and use them as variable names.

How are you feeling?

slack

🤯

I have no idea what you're talking about

✋

I have some questions but feel like I understand some things

👍

I feel comfortable with everything you've said

Rules for variable names

Variable names are also case-sensitive, so `myvariable` is not the same as `MyVariable` or `MYVARIABLE`

Rules for variable names

Write this in a cell:

```
myvariable = “I love python”

MyVariable = “ I hate python”

MYVARIABLE = “I am learning python”

print(myvariable)

print(MyVariable)

print(MYVARIABLE)
```

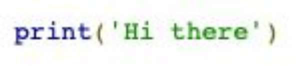
What do you notice?

Demo

Topic 4: Printing

Printing

- Here is a simple example:

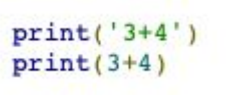


```
print('Hi there')
```

- The print function requires parenthesis around its arguments. In the program above, its only argument is the string 'Hi there'.
- Anything inside quotes will be printed exactly as it appears. But there are a few exceptions....

Printing

- Try the following. What is the difference?

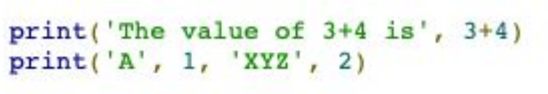


```
print('3+4')
print(3+4)
```

- The first statement will output 3+4, while the second will output 7.

Printing

- To print several things at once, separate them by commas. Python will automatically insert spaces between them. Below is an example and the output it produces.



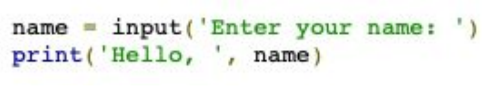
```
print('The value of 3+4 is', 3+4)
print('A', 1, 'XYZ', 2)
```

- What do you see?

Topic 5: Inputs

Getting Input

- The input function is a simple way for your program to get information from people using your program. Here is an example:



```
name = input('Enter your name: ')
print('Hello, ', name)
```

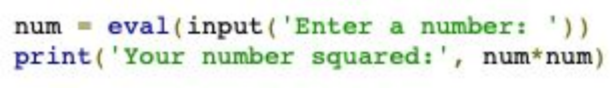
- The basic structure is

<variable name> = input(*<message to user>*)

- This works for getting text from the user. What about getting numbers?

Getting Input

- To get numbers from the user to use in calculations, we need to do something extra. Here is an example:



```
num = eval(input('Enter a number: '))
print('Your number squared:', num*num)
```

- The eval function converts the text entered by the user into a number. One nice feature of this is you can enter expressions, like 3*12+5, and eval will compute them for you.

Task 1:

Write a program that asks for the users name then prints ‘Welcome [name]’

Let’s practice!

Task 2:

The Body Mass Index, BMI, is calculated as
BMI = 703w/h(^2)

where w is the person’s weight in pounds and h is the person’s height in inches.
Write a program that asks the user for their height their weight and prints out their BMI. [Note: one way to compute h(^2) is as h * h.]

Task 3:

Write a program that asks the user to enter their name. Then print out the user’s name three times on the same line.

Session 2

RECAP

-Basic Python Syntax
- Indentation
- Variables(case-sensitivity)
- Global and local scoping
- Python comment
- Casting

-Python Data Types

-Python Arithmetic operations
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators

Task 4:

Write a program that asks the user to enter a number. Store that number in a variable. Add 2 to that number, store the result in the same variable, and then print out the value of that variable

Task 5:

Write a program that asks the user to enter five numbers (use five input statements). Then print out those numbers all on the same line, with each number separated from the others by exactly three spaces. Research into the ‘sep’ optional argument to the print statement to do this.

BLACK DISRUPTOR ACCELERATOR

Task 6:

Create a login system.
The user should input their name, and age. Print out the users age in Dog Years, in the following format:
‘Hello [name], you are [age in dog years] years old, in Dog years’
eg. Hello John, you are 140 years old, in Dog Years.

BLACK DISRUPTOR ACCELERATOR

Task 7:

Write a program to calculate the area of a football field and its perimeter?

PYTHON LIST METHODS

BLACK DISRUPTOR ACCELERATOR

append
copy
index
insert
pop
remove
reverse
sort

Thank You