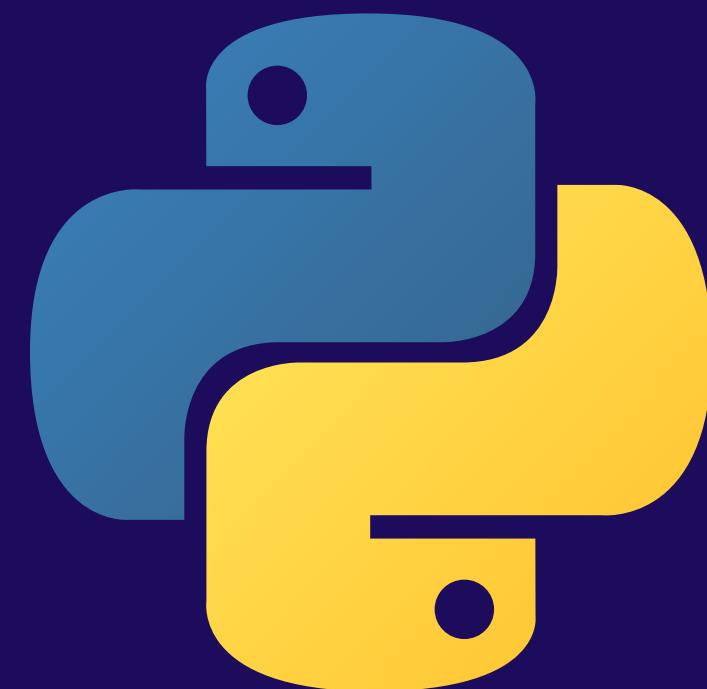




WELCOME TO PYTHON CLASS DAY 9



DATA TYPES

Data types in Python are categories that tell us what kind of information something is. They help the computer know how to store and work with different kinds of information.



TOMMY TREASURE BOX

It's Story Time

cave of wonders



Integers

1,2,-3,0



In Python, an integer is a **whole number**, either positive, negative, or zero, without any decimal points or fractions, represented as a single value, like **1, -5, or 0.**

Integers (int)

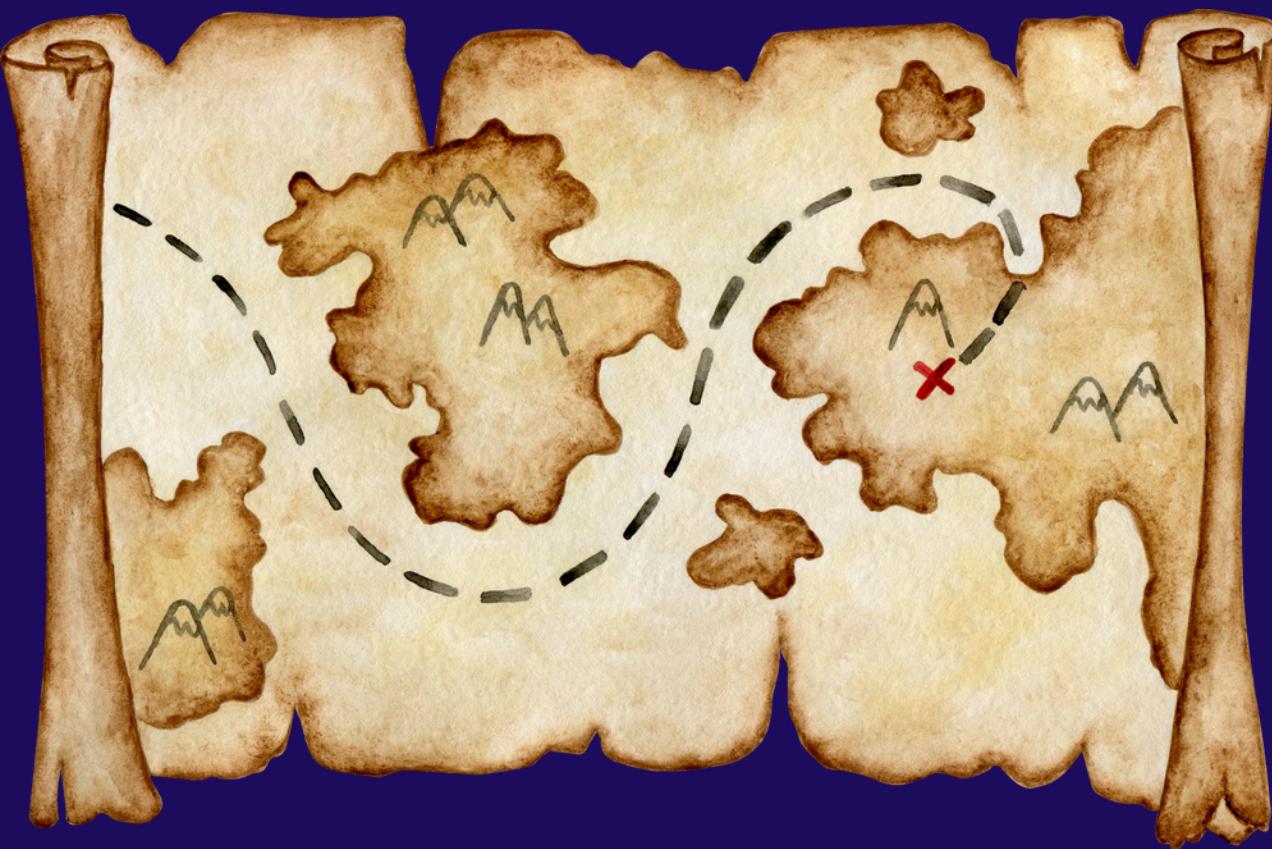
coins = 10

print(coins)

10

Float

3.15km,
2.2km,
0.9km



In Python, a float is a number that contains a fractional part, represented as a single value with a decimal point, such as 3.14, -0.5, or 2.0. Floats can be positive, negative, or zero, and are used to represent numbers that require accuracy in calculations, like measurements, percentages."

Floats (float)

map = 3.14

print(map)

3.14

strings



"In Python, a string is a sequence of characters, such as letters, numbers, or symbols, enclosed in quotes .

Examples of strings include 'hello',
"goodbye", or '123abc'."

```
# Strings (str)
```

```
scroll = "Hello, Tommy!"
```

```
print(scroll)
```

Boolean

TRUE
FALSE



```
# Boolean (bool)  
secret_code = True  
print(secret_code)
```

True

"In Python, a boolean is a logical value that can have only two possible values: **True** or **False**. Boolean values are typically used in if-else statements, loops to make decisions"

lists



[car, controller, dice]

In Python, a list is a collection of items that can be of **any data type**, including strings, integers, floats, accessed and manipulated using their **index position**. Examples of lists include **[1, 2, 3]**, **["a", "b", "c"]**, **[1, "a", 3.14]**

Lists (list)

```
toys = ["Car", "Doll", "Ball"]
```

```
print(toys)
```

```
["Car", "Doll", "Ball"]
```

Dictionaries



“golden sword”: “dragon cave”,

“rare coin”: “ancient building”

"In Python, a dictionary (dict) is an
unordered collection of **key-value pairs**
Defined by curly brackets {} and key-
value pairs separated by commas, like
`{'name': 'John', 'age': 30}.`"

```
# Dictionaries (dict)
```

```
treasure_lock = {"key1": "Treasure1",  
"key2": "Treasure2"}
```

```
print(treasure_lock)
```

```
{"key1": "Treasure1", "key2": "Treasure2"}
```

Data Types

Name	Type	Description
Integers	int	Whole numbers, such as: 3 300 200
Floating point	float	Numbers with a decimal point: 2.3 4.6 100.0
Strings	str	Ordered sequence of characters: "hello" 'Sammy' "2000" "楽しい"
Lists	list	Ordered sequence of objects: [10,"hello",200.3]
Dictionaries	dict	Unordered Key:Value pairs: {"mykey": "value", "name": "Frankie"}
Tuples	tup	Ordered immutable sequence of objects: (10,"hello",200.3)
Sets	set	Unordered collection of unique objects: {"a","b"}
Booleans	bool	Logical value indicating True or False

Python is dynamically-typed, which means it can often automatically infer the data type of a variable based on the value assigned to it, without needing explicit type declaration. For example:

x = 5 # Python knows x is an integer

y = 3.14 # Python knows y is a float

name = "John" # Python knows name is a string

