Intro to Coding in Python

Lesson 1: Python Basics

Brought to you by the University of Maryland
Balloon Payload Program

How do Computers Operate?

Digital: Uses numbers (digits)

All computers store and read information as 1's and 0's

- Numbers, letters, pictures, instructions, etc.
- We call this **Binary**

Controlling this information lets us control the computer

Binary Converter

Char.	ASCII	Char.	ASCII	Char.	ASCII
@	64	U	85	j	106
A	65	V	86	k	107
B	66	W	87	l	108
C	67	X	88	m	109
D	68	Y	89	n	110
E	69	Z	90	o	111
F	70	[	91	p	112
G	71	\	92	q	113
H	72	]	93	r	114
I	73	^	94	s	115
J	74	_	95	t	116
K	75	`	96	u	117
L	76	a	97	v	118
M	77	b	98	w	119
N	78	c	99	x	120
O	79	d	100	y	121
P	80	e	101	z	122
Q	81	f	102	{	123
R	82	g	103	|	124
S	83	h	104	}	125
T	84	i	105	~	126

Character **Decimal** **Binary**

B → 66 → 0100010

Example: text to binary

What is a Coding Language?

Just like we speak different languages, so do computers

- Different syntax
- Intended purposes/features (C++, Assembly, Python, etc)
- Compiled vs interpreted

The middle ground between **human instructions** and **binary**

- A list of commands you can give your computer
  Like english is a list of words you can give to a person

How do we use a Coding Language?

Programming: the process of writing code to give computers instruction (perform tasks or solve problems)

Must understand how a system **Processes Data**

Input → Logic → Output

Can do this through different programming applications (python has many options)

We will use a browser based IDE

https://spacepython.com/en/editor/

Find This and Past Lessons:

Add link

Data Types

In a coding language, you need to work with **data**. Data comes in different forms calls **types**.

Integer: 1, 2, 3, 4, 5, …
Float: 1.5, 20.2, 100.12354, 3.1415, …
Character: ‘a’, ‘b’, ‘c’, …
String: ‘Hello World’, ‘abcdefg’, …
Boolean: True, False

We treat data types differently

Type Practice!

7.4 ----------> **float**

1.2 ----------> **float**

1000000000 ----------> **integer**

-12 ----------> **integer**

False ----------> **boolean**

“False” ----------> **string**

“4.5” ----------> **string**

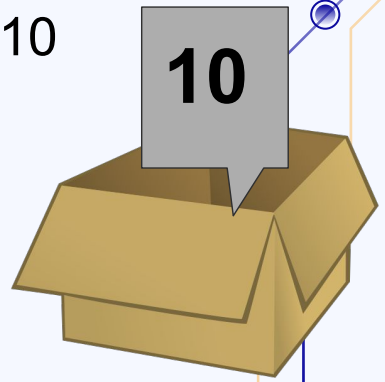
[1,2,3] ----------> **list**

(1,2,3) ----------> **tuple**

What is a Variable?

A variable can take any **data**, and have any **name**
It's like a labeled box that holds whatever you put into it.

X = 10 Variable called "X", with integer value 10
word = "Hello" Variable called "Word", with String value "Hello"
OR:
X = "Hello" Variable called "X", with string value "Hello"
word = 10 Variable called "word", with integer value 10



Initializing Variables

When you create a variable, Python will assign it a type. In other programming languages, you have to specify the type yourself.

Int: x = 20

Float: x = 20.5

String: x = “Hello World” OR x = ‘Hello World’

List: x = ["apple", "banana", "cherry"]

Boolean: x = True

Print Statements

Syntax: *print(message)*

Example:

Print strings: *print("Hello World")*
Print numbers: *print(10)*

The print statement **displays** messages in the **console** (area on the right).

User Inputs

Syntax: *input(message)*

Example (try it yourself):

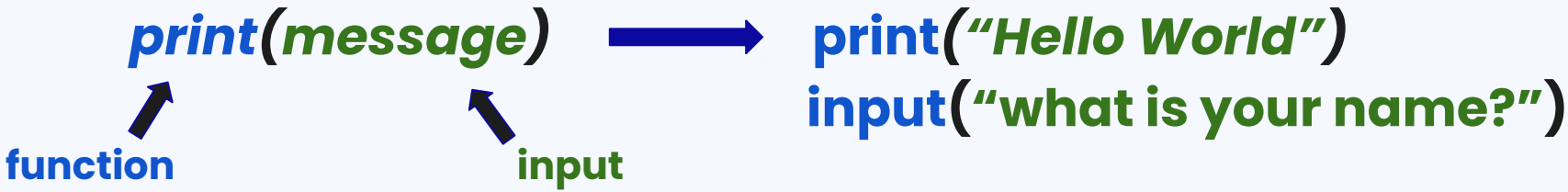
```
1  x = input("what is your name?")
2  print(x)
```

Input is like a print statement, but will wait for the **user** to type in **input** to the **console** before continuing

Calling Functions

Syntax: *function(inputs)*

Example:



Python has many built in functions you can use for convenience (one google search away)

Commenting your Code

Syntax: *#comment*

```
print('hello world') # prints Hello World to the console

# in python anything you write after the '#' is considered
# a comment and will be skipped when the computer executes
# your code

# comment a single line with #

'''
or comment entire portions of code using """ on either side
of the block you want to comment

'''
```

Easy Mathematical Operators

Addition:	+
Subtraction:	-
Multiplication:	*
Division:	/
Exponentiation:	**

Keep in mind what **type** of data you are operating on

int + int = int

int + float = float

You cannot operate with incompatible data types!

Addition vs Concatenation

1.
```
print('hello' + 'world')
```

2.
```
print(9+10)
```

3.
```
print('hello' + 10)
```

What happens for each one?

Conditional Statements

What if you need your computer to make simple decisions for you?

3 different statements: **if, else if, else**

If: if conditional is true/untrue, then execute a code block

Else if: if the previous conditional is not met BUT a new condition is met, execute a code block

Else: if ALL previous conditionals are NOT met, execute a code block

Example (try it yourself)

```
1  x = 10 #change this number and see what happens
2
3  if x < 5:
4      print('x is less than 5')
5  elif x < 10:
6      print('x is less than 10')
7  else:
8      print('x is not less than 5 or 10')
```

Whitespace in Python

Python specifically uses **indentation** as part of it's **syntax**

Indentation defines **code blocks**
Blank lines are simply skipped

Notice the difference in indentation within a conditional statement

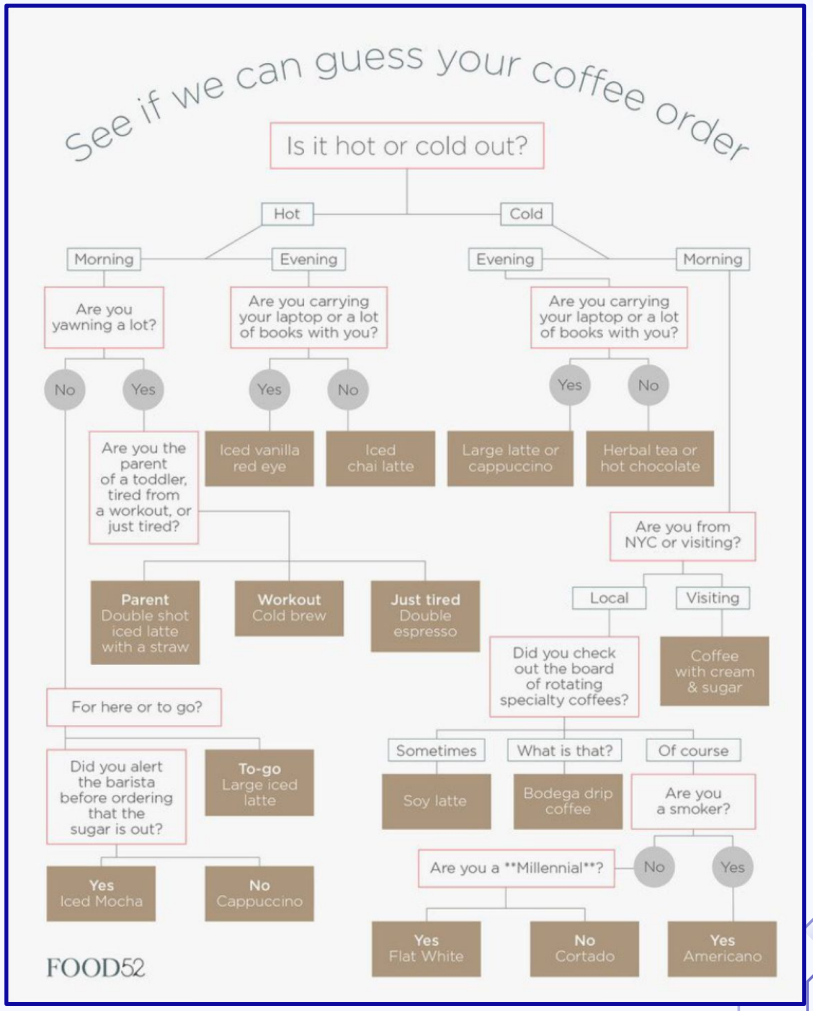
```
if x < 5:
    print('x is less than 5')
elif x < 10:
    print('x is less than 10')
else:
    print('x is not less than 5 or 10')
```

Exercise time!

Let's practice your new knowledge

Write your own **Decision Tree**

- Ask a series of questions before coming to a different conclusion depending on how the user answers
- You can follow the given diagram or write your own!
- Other ideas: can I guess what pet you have, guess your favorite color, guess which state you live, etc.



Short example

```
print("Welcome! Can I guess your Coffee Order?")
temp = input('Is is hot or cold out?')

if temp == 'hot':
    tired = input('is to evening or morning?')

    if tired == 'evening':
        action = input('are you carrying a lot of books with you?
(yes/no)')

    if tired == 'morning':
        action = input('are you yawning a lot? (yes/no)')

        if action == 'yes':
            order = input('for here or to go? (here/go)')

            if order == 'go':
                final_ans = 'Large Iced Latte'

print('Your order is a ' + final_ans)
```