



Cognitio Learning

Day 1 Python

Day 1 covers the basics of Python.

Printing

The basics of any programming language include learning how to print something to your console. In Python, you accomplish this by running `print('Hello World')` .

```
print('Hello World') # This is a comment
print('Hello World') # This is another comment

content = 'Hello World'
print(content) # Hello World
```

The `print` function takes the `'Hello World'` input and outputs it to the console.

Variables

Variables are a way to store information to reuse later on. The standard convention for variable naming in Python is `snake_case` .

```
some_number = 10
some_number_string = str(some_number) # '10' - converts the number into a string format
```

Now you might be asking yourself, what are strings? numbers? The next section covers what these really are.

Data Types

Data types are the classification of data items.

Strings

Strings are sequences of characters that form text.

```
name = "Alice"
greeting = 'Hello'
print(name) # Alice
print(greeting) # Hello
```

Numbers

Integers is a whole number.

Floats are numbers with a decimal point.

```
age = 30 # Integer
pi = 3.14 # Float
```

Booleans

Booleans represent either `True` or `False`

```
is_sunny = True
is_raining = False
print(is_sunny) # True
print(is_raining) # False
```

Lists

Lists are a collections of items.

```
fruits = ['apple', 'banana', 'cherry']
print(fruits) # ['apple', 'banana', 'cherry']
```

Dictionaries

Dictionaries are collections of key-value pairs.

```
person = {
```

```
'name': 'Alice',
'age': 30,
'is_student': False
}
print(person)  # {'name': 'Alice', 'age': 30, 'is_student': False}
```

Conditional Statements

Conditional statements allow you to execute different code based on certain conditions.

```
apples = 5

if apples < 5:
    print("You have less than 5 apples.")
elif apples == 5:
    print("You have 5 apples.")
else:
    print("You have more than 5 apples.")
```

Here we define the variable `apples` as `5`. We have a conditional statement (if statement) to check if the value of `apples` is less than 5, as shown by `if apples < 5`. Because the value of `apples` is not less than 5, it moves on.

Note that our next conditional statement is `elif` which means else if. Else statements trigger if the original conditional statement is false. In this case, because `apples` is not less than 5, it runs.

Our next conditional statement is checking if `apples ==` and it is! Thus, the code would output `You have 5 apples.` However, if `apples` was 6 what would happen? We don't have any code that checks for if `apples > 5` right?

Although this is true, we checked for if `apples` was less than 5 and if `apples` is equal to 5. Meaning that if both of those are false, then the only thing left is `apples` being greater than 5.

Exercise

Experiment with changing the value of `apples` to be different values. See how the output changes.

Now, make a conditional statement to check the age of someone to see if they are a child or an adult. (If they are 18 or older, they are an adult)

Loops

Loops allow you to execute a block of code multiple times.

For Loops

For loops iterate over a sequence (like a list or a string).

```
fruits = ['apple', 'banana', 'cherry']

for fruit in fruits:
    print(fruit)
# apple
# banana
# cherry
```

Let's say you like playing video games and watching TV, and I wanted you to say all the things you like. You would say I like playing video games and I like watching TV . Notice how you **looped** over the things you liked and then said I like ____ and replaced the blank with the activity you like.

That is essentially a for loop! In the code above, we are doing the same thing except we are going through the list of fruits and printing each fruit.

While Loops

While loops repeat as long as a condition is true.

```
count = 0

while count < 5:
    print(count) # prints value of 'count' to the console
    count += 1 # this increases the value of count BY 1
# 0
# 1
# 2
# 3
# 4
```

Imagine you are eating candy in your basket until it's empty, and you don't know how many candies there are in the basket. You would keep eating until there were no more. The same thing is happening here, except the condition to which you stop isn't limited to just running out.

In the code above, we run through the code in the function once, printing 0 and increasing count

to 1 . We run through the code, printing 1 and increasing count to 2 . We keep doing this until count is at 4 . The next time we run this, we print 4 and we increase count by 5 . However, the loop ends now because count (5) is no longer less than 5 .

Conclusion

This concludes the basic overview of Python for Day 1. With these fundamentals, we'll continue onto day 2 with more advanced concepts.