

Python Basics

How much have I forgotten ?

Standard arithmetic operations

1. $2+3*4$ 14
2. $(2+3)*4$ 20
3. $2**10$ 1024
4. $6/3$ 2.0
5. $7/3$ 2.333333333333
6. $7//3$ 2
7. $7\%3$ 1
8. $3/6$ 0.5
9. $3//6$ 0
10. $3\%6$ 3
11. $2**100$ 1267650600228229401496703205376

Comparison operators

1. `5 == 10` `False`
2. `10 > 5` `True`
3. `(5 >= 1) and (5 <= 10)` `True`

Variable

```
>>> the_sum = 0
```

```
>>> the_sum
```

0

```
>>> the_sum = the_sum + 1
```

```
>>> the_sum
```

1

```
>>> the_sum = True
```

```
>>> the_sum
```

True



Figure 1.3: Variables Hold References to Data Objects

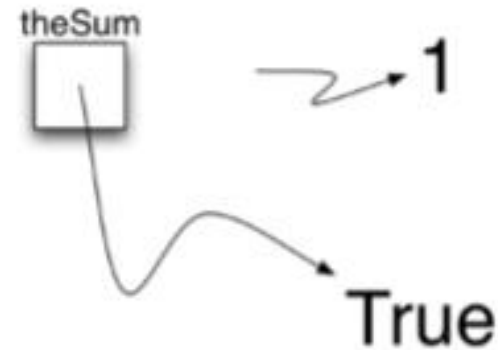


Figure 1.4: Assignment changes the Reference

Built-in Collection

```
>>> [1,3,True,6.5]
```

```
[1, 3, True, 6.5]
```

```
>>> my_list = [1,3,True,6.5]
```

```
>>> my_list
```

```
[1, 3, True, 6.5]
```

repetition

```
>>> my_list = [0] * 6
```

```
>>> my_list
```

```
[0, 0, 0, 0, 0, 0]
```

references

```
>>> my_list = [1,2,3,4]
>>> A = [my_list]*3
>>> print(A)
[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
>>> my_list[2]=45
>>> print(A)
[[1, 2, 45, 4], [1, 2, 45, 4], [1, 2, 45, 4]]
>>> print(my_list)
[1, 2, 45, 4]
```

The variable A holds a collection of three references to the original list called my_list.

Note that a change to one element of my_list shows up in all three occurrences in A.

```
my_list = [1024, 3, True, 6.5]
my_list.append(False)
print(my_list)
my_list.insert(2,4.5)
print(my_list)
print(my_list.pop())
print(my_list)
print(my_list.pop(1))
print(my_list)
my_list.pop(2)
print(my_list)
```

```
my_list.sort()
print(my_list)
my_list.reverse()
print(my_list)
print(my_list.count(6.5))
print(my_list.index(4.5))
my_list.remove(6.5)
print(my_list)
del my_list[0]
print(my_list)
```



```
>>> my_list = [1024, 3, True, 6.5]
>>> my_list.append(False)
>>> print(my_list)
[1024, 3, True, 6.5, False]
>>> my_list.insert(2,4.5)
>>> print(my_list)
[1024, 3, 4.5, True, 6.5, False]
>>> print(my_list.pop())
False
>>> print(my_list)
[1024, 3, 4.5, True, 6.5]
>>> print(my_list.pop(1))
3
>>> print(my_list)
[1024, 4.5, True, 6.5]
>>> my_list.pop(2)
True
```

```
>>> print(my_list)
[1024, 4.5, 6.5]
>>> my_list.sort()
>>> print(my_list)
[4.5, 6.5, 1024]
>>> my_list.reverse()
>>> print(my_list)
[1024, 6.5, 4.5]
>>> print(my_list.count(6.5))
1
>>> print(my_list.index(4.5))
2
>>> my_list.remove(6.5)
>>> print(my_list)
[1024, 4.5]
>>> del my_list[0]
>>> print(my_list)
[4.5]
```

list & range

```
>>> range(10)
```

```
range(0, 10)
```

```
>>> list(range(10))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> range(5,10)
```

```
range(5, 10)
```

```
>>> list(range(5,10))
```

```
[5, 6, 7, 8, 9]
```

```
>>> list(range(5,10,2))
```

```
[5, 7, 9]
```

```
>>> list(range(10,1,-1))
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2]
```

Exercise 1 (Input)

- Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.
- Add on to your program asking the user for another number and printing out that many copies of the message on separate lines.

Exercise 2 (Rock-Paper-Scissors)

- Make a two-player Rock-Paper-Scissors game.
- Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game
- Remember the rules:
 - Rock beats scissors
 - Scissors beats paper
 - Paper beats rock

Exercise 3 (Fibonacci sequence)

Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate.

(Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, ...)

Exercise 4 (Password Generator)

Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password. Include your run-time code in a main method.

Ask the user how strong they want their password to be. For weak passwords, pick a word or two from a list.

Python's random module (see Python documentation)

You've got to defuse a bomb by placing exactly 4 gallons of water on a sensor. The problem is, you only have a 5 gallon (18.9 L) jug and a 3 gallon jug on hand! This classic riddle, made famous in Die Hard 3, may seem impossible without a measuring cup, but it is actually remarkably simple.

https://www.youtube.com/watch?v=BVtQNK_ZUJg



Prisoner Hat

<https://www.youtube.com/watch?v=N5vJSNXPEwA>

Three Gods Riddle

<https://www.youtube.com/watch?v=LKvjlsyYng8>

Goat !!

<https://www.youtube.com/watch?v=mhlc7peGlGg>

Integer Length

Write a function that takes an integer as input and returns the number of digits in that integer.

https://adriann.github.io/programming_problems.html

Work on

- 2 problems from Elementary section
- 2 problems from Lists, Strings section
- 1 problem from Intermediate or Advanced sections

Coding for Kids

KIBO

- <https://www.youtube.com/watch?v=baod8Lg3iCQ>
- https://youtu.be/lr0iUK_apdM?list=PLsXbKpJZTa6OEDWJ4ZQkT0CFE4BT8KkAD
- <https://youtu.be/F7pbknI5JOI?list=PLsXbKpJZTa6OEDWJ4ZQkT0CFE4BT8KkAD>

Coding for Kids

- Cubetto
- https://youtu.be/NzF_ng0ELZw
- Osmo
- <https://youtu.be/iosNyOslcPY>
- <https://www.youtube.com/watch?v=jn56Dm6Ozbg>

- Lightbot

<https://www.youtube.com/watch?v=VLI0glygHx8>

- Kodable

<https://www.youtube.com/watch?v=wazj9WBdvjw>

Integer Length

Write a function that takes an integer as input and returns the number of digits in that integer.

By applying aforementioned steps, we can come up with this pseudo code:

1. Get the number
2. Divide it by 10, if the result is more than 1 that the number is greater than 10
3. Increment the counter (we now know that we have at least one if the result is greater than 1)
4. Repeat division with the result as a new number until new result is less than 1
5. Exit by returning the counter value