

Programming in Python

Python

- Easy to learn!
- Extremely useful, widely used

Downloading / Installing Python

- We'll be using IDLE
- Check – might already be installed on computers (search IDLE in Windows)
- <https://www.python.org/downloads/>



Download Python 2.7.14

- Install – ask me if you need help!

First Program: Hello World!

- “Print” a message

```
print(“Hello World”)
```

← Use “quotations” to print text

To run program: click on



Variables

- Something that stores a value
- We can change these values and use them to store information

```
print("Justin")
```

```
name = "Justin"  
print(name)
```

```
x = 2  
print(x)  
y = 3  
print(y)
```

```
sum = x + y  
print(sum)
```

If / else statements

- Used to do some task if a statement is a true
- Example: if $x > 5$, then print “x is greater than 5”

```
x = 3
```

```
if x > 5:
```

```
    print(“x is bigger than 5”)
```

```
else:
```

```
    print(“x is smaller than 5”)
```

Try it Yourself: Enter a number and print if it is positive or negative

Hint (obvious): all positive numbers are greater than 0, all negative numbers are less than 0

You can name your variable anything!

Solution

```
number = -10
```

```
if number > 0:  
    print("This number is positive")  
elif number < 0:  
    print("This number is negative")  
else:  
    print("This number is zero!")
```


While loop

- While some statement is true, do this task
- When it is no longer true, stop doing that task

```
while True:  
    print("loop")
```

*This will run forever!
Careful!*

```
while 1 < 2:  
    print("infinite loop!")
```

```
x = 0  
while x < 3:  
    x = x + 1  
    print(x)
```

This will print:

*1
2
3*

For Loops

- Loop over a specified range
- Good because no infinite loops!

```
for number in range(0,5):  
    print(number)
```

*”number” is a variable that is
automatically created, is
automatically set to 0*

This will print:

0

1

2

3

4

Challenge

- Input: a number greater than 0
- Output: all even numbers between 0 and that number
- Hint: use a loop
- Another hint: `range(1, 10, 2)` gives you every other number from 1 to 10

Solution

```
input = 11  
k = 0  
while k < input:  
    print(k)  
    k += 2
```

OR

```
for i in range(0, input, 2):  
    print(i)
```

These are all correct!
There are many ways to solve this problem!

Any questions?

Functions

- Create one piece of code that we can use many times

```
def function_name(input):  
    do something  
    return answer
```

```
def add_one(number):  
    answer = number + 1  
    return answer
```

Functions

```
def add_one(number):  
    answer = number + 1  
    return answer  
  
number_plus_one = add_one(4)  
print(number_plus_one)
```

What's happening here?

When we run this code:

1. Python sees that we defined a function “`add_one`”
2. Python remembers this, and moves on
3. Python creates a variable “`number_plus_one`”
4. “`number_plus_one`” uses the “`add_one`” function, so Python runs the “`add_one`” function with “4” as an input”
5. “`add_one(4)`” returns a value of 5
6. “`number_plus_one`” = 5
7. Python prints “`number_plus_one`”, which is 5!

Try it yourself

- Write a function that takes prints your name and a message
- Example:
 - Input: "Justin"
 - Output: "Hello Justin"
- Hint: to print multiple things, use a comma or a "+" to separate strings
- Example:
 - `print("Hello Justin")`
 - `print("Hello", "Justin")`
 - `print("Hello " + "Justin")`

Solution

```
def say_hello(name):  
    print("Hello", name + "!")  
say_hello("Justin")
```

Use "input" to write an interactive program!

"input" is a function included in Python!

```
name = raw_input("Please enter your name: ")  
say_hello(name)
```


Lists

```
my_first_list = [1,2,3,4]
```

```
one = list[0]
```

```
two = list[1]
```

```
four = list[3]
```

```
print(one)
```

```
print(two)
```