

Intro to Programming

What is programming?



Programming is not a bunch of crazy numbers!

It's giving computers a set of instructions!



A Special Language

A language to talk
to dogs!



Programming is a
language to talk to
computers

People are smart! Computers are dumb!

SALAD INSTRUCTIONS

Programming is like a recipe!

Computers do EXACTLY what you say, every time.

Which is great if you give them a good recipe!

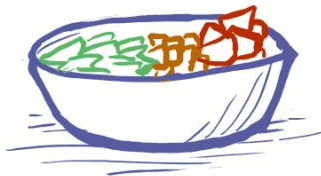
1) GET A LETTUCE HEAD, A CARROT, A TOMATO, A KNIFE, AND A BOWL



2) USE THE KNIFE TO CUT UP THE LETTUCE HEAD, CARROT, AND TOMATO



3) PUT THE LETTUCE, CARROT AND TOMATO IN THE BOWL



4) MIX THE CONTENTS OF THE BOWL



People are smart! Computers are dumb!

But if you get it
out of order....

A computer
wouldn't know
this recipe was
wrong!

SALAD INSTRUCTIONS

1) GET A LETTUCE HEAD,
A CARROT, A TOMATO, A
KNIFE, AND A BOWL



3) PUT THE LETTUCE,
CARROT AND TOMATO
IN THE BOWL



2) USE THE KNIFE TO CUT
UP THE LETTUCE HEAD,
CARROT, AND TOMATO



4) MIX THE CONTENTS
OF THE BOWL



People are smart! Computers are dumb!

Computers are bad at filling in the gaps!

A computer wouldn't know something was missing, it would just freak out!

SALAD INSTRUCTIONS



Everyone/thing has strengths!



- Understand instructions despite:
 - Spelling mistakes
 - Typos
 - Confusing parts
- Solve problems
- Tell computers what to do
- Get smarter every day



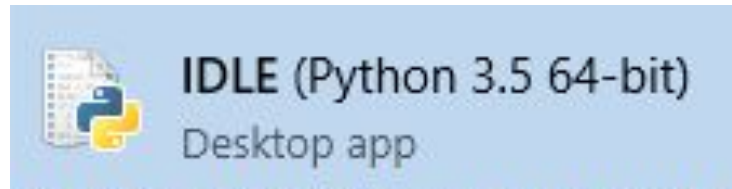
- Does exactly what you tell it
- Does it the same every time
- Doesn't need to sleep
- Will work for hours on end
- Doesn't get bored
- Really really fast
- Get smarter when you tell it how

Intro to Python

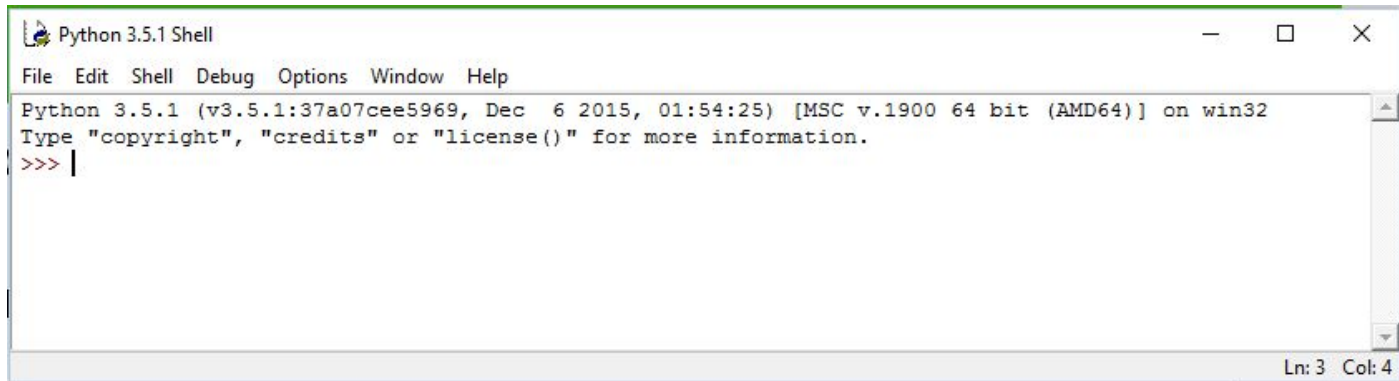
Let's get coding!

Where do we program? In IDLE

Click the start button and type IDLE!



Make sure the first number after "Python" is 3!



Make a mistake!

Type by **button mashing** the keyboard!

Then press enter!

asdf asdjlkj;pa j;k4uroei

Did you get a big red error message?

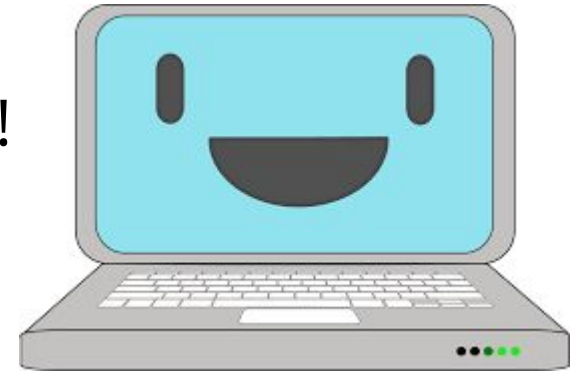
Mistakes are great!

*SyntaxError:
Invalid Syntax*

Good work you made an error!

*ImportError:
No module
named humour*

- Programmers make A LOT of errors!
- Errors give us hints to find mistakes
- Run your code often to get the hints!!
- Mistakes won't break computers!



*KeyError:
'Hairy Potter'*

*AttributeError:
'NoneType' object
has no attribute
'foo'*

*TypeError: Can't
convert 'int' object
to str implicitly*

We can learn from our mistakes!

Error messages help us fix our mistakes!
We read error messages from bottom to top

Traceback (most recent call last):

File "C:/Users/Madeleine/Desktop/tmp.py", line 9, in <module>
 print("I have " + 5 + " apples")

TypeError: can only concatenate str (not "int") to str

1. What went wrong

2. What code didn't work

3. Where that code is

Write some code!!

Type this into the window
Then press enter!

```
print('hello world')
```

Did it print:

hello world

???

Tell me more!

We can `print` things in lots of different ways in python!

```
>>> print("Hello world!")
```

```
>>> print("Hello", "world!")
```

```
>>> print("Hello", "world", end="!")
```

Tell me more!

We can `print` things in lots of different ways in python!

```
>>> print("Hello world!")
```

```
Hello world!
```

```
>>> print("Hello", "world!")
```

```
>>> print("Hello", "world", end="!")
```

Tell me more!

We can `print` things in lots of different ways in python!

```
>>> print("Hello world!")
```

```
Hello world!
```

```
>>> print("Hello", "world!")
```

```
Hello world!
```

```
>>> print("Hello", "world", end="!")
```


Tell me more!

We can `print` things in lots of different ways in python!

```
>>> print("Hello world!")
```

```
Hello world!
```

```
>>> print("Hello", "world!")
```

```
Hello world!
```

```
>>> print("Hello", "world", end="!")
```

```
Hello world!
```

Note that this last one will not have a new line after it!

Tell me more!

We can `print` on many lines at once!

```
>>> print("""Hello world.
```

```
This is me!
```

```
Life should be fun for everyone""")
```

Tell me more!

We can `print` on many lines at once!

```
>>> print("""Hello world.
```

```
This is me!
```

```
Life should be fun for everyone""")
```

```
Hello world.
```

```
This is me!
```

```
Life should be fun for everyone
```

Python the calculator!

Try writing some maths into python!

```
>>> 1 + 5
```

```
>>> 2 - 7
```

```
>>> 2 * 8
```

```
>>> 12/3
```

Python the calculator!

Try writing some maths into python!

```
>>> 1 + 5
```

```
6
```

```
>>> 2 - 7
```

```
>>> 2 * 8
```

```
>>> 12/3
```

Python the calculator!

Try writing some maths into python!

```
>>> 1 + 5
```

```
6
```

```
>>> 2 - 7
```

```
-5
```

```
>>> 2 * 8
```

```
>>> 12/3
```

Python the calculator!

Try writing some maths into python!

```
>>> 1 + 5
```

```
6
```

```
>>> 2 - 7
```

```
-5
```

```
>>> 2 * 8
```

```
16
```

```
>>> 12/3
```

Python the calculator!

Try writing some maths into python!

```
>>> 1 + 5
```

```
6
```

```
>>> 2 - 7
```

```
-5
```

```
>>> 2 * 8
```

```
16
```

```
>>> 12/3
```

```
4
```


A calculator for words!

What do you think these bits of code do?

Try them and see!

```
>>> "cat" + "dog"
```

```
>>> "tortoise" * 3
```

A calculator for words!

What do you think these bits of code do?

Try them and see!

```
>>> "cat" + "dog"
```

```
catdog
```

```
>>> "tortoise" * 3
```

A calculator for words!

What do you think these bits of code do?

Try them and see!

```
>>> "cat" + "dog"
```

```
catdog
```

```
>>> "tortoise" * 3
```

```
tortoisetortoisetortoise
```

Strings!

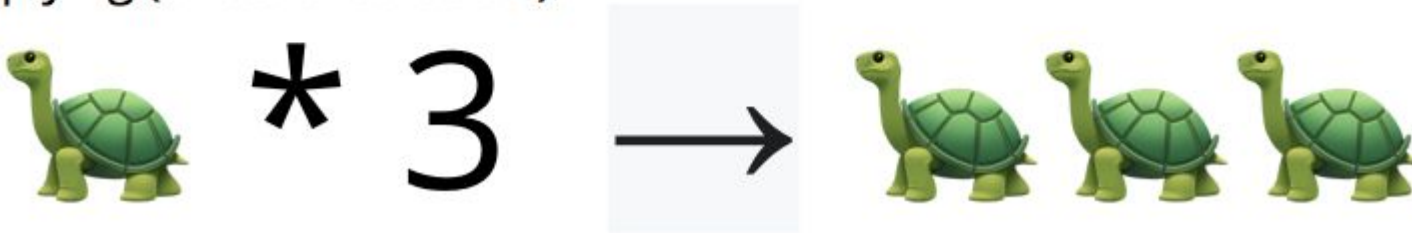
Strings are things with "quotes"

To python they are essentially just a bunch of pictures!

Adding :



Multiplying (3 lots of tortoise!):



Strings!

Strings can have any letters in them, even just spaces!

```
"Hello, world!"
```

```
"bla bla bla"
```

```
":)"
```

```
" "
```

```
'I can use single quotes too!'
```

```
"~\_(\ツ)\_/~"
```

```
"asdfghjklqwertyuiopzxcvbnm"
```

```
"DOGS ARE AWESOME!"
```

```
"!@#$%^&*()_+--[|\\:;'<>,./?"
```

Strings and Ints!

Integers are numbers in python.

We can do maths with integers but not strings

```
>>> 5 + "5"
```

We can turn a string into an integer using `int()`

```
>>> 5 + int("5")
```

Similarly, we turn an integer into a string using `str()`

```
>>> str(5) + "5"
```

Strings and Ints!

Integers are numbers in python.

We can do maths with integers but not strings

```
>>> 5 + "5"
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

We can turn a string into an integer using `int()`

```
>>> 5 + int("5")
```

Similarly, we turn an integer into a string using `str()`

```
>>> str(5) + "5"
```

Strings and Ints!

Integers are numbers in python.

We can do maths with integers but not strings

```
>>> 5 + "5"
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

We can turn a string into an integer using `int()`

```
>>> 5 + int("5")
```

```
10
```

Similarly, we turn an integer into a string using `str()`

```
>>> str(5) + "5"
```


Strings and Ints!

Integers are numbers in python.

We can do maths with integers but not strings

```
>>> 5 + "5"
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

We can turn a string into an integer using `int()`

```
>>> 5 + int("5")
```

```
10
```

Similarly, we turn an integer into a string using `str()`

```
>>> str(5) + "5"
```

```
'55'
```

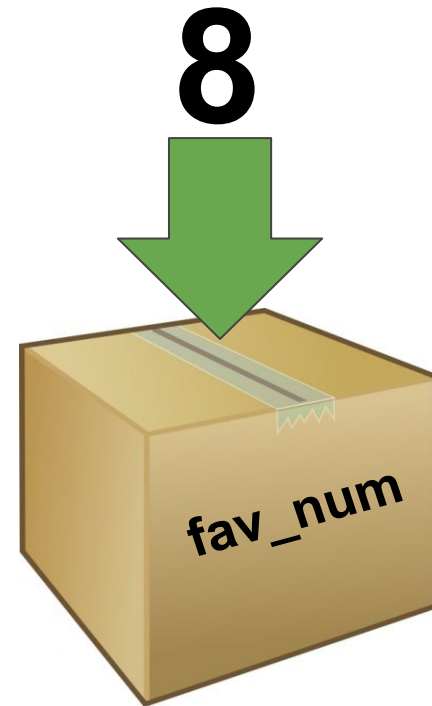
No Storing is Boring!

It's useful to be able to remember things for later!

Computers remember things in "**variables**"

Variables are like putting things into a **labeled cardboard box**.

Let's make our favourite number 8 today!



Variables

Instead of writing the number 8, we can write fav_num.



$$\text{fav_num} - 6 \\ \Rightarrow 2$$

$$\text{fav_num} + 21 \\ \Rightarrow 29$$

$$\text{fav_num} * 2 \\ \Rightarrow 16$$

$$\text{fav_num} / 2 \\ \Rightarrow 4$$

Variables

Instead of writing the number 8, we can write fav_num.



fav_num - 6
=> 2

fav_num + 21
=> 29

fav_num * 2
=> 16

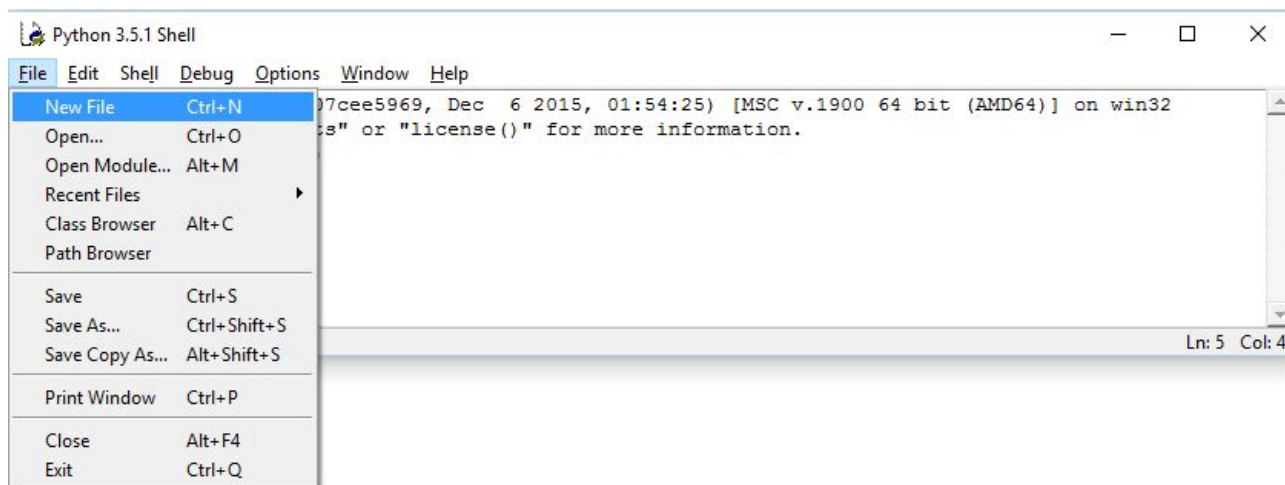
We'll come back to this later!

But writing 8 is
much shorter than
writing fav_num???



Coding in a file!

Code in a file is code we can run multiple times! Make a reusable “hello world”!



1. Make a new file called hello.py, like the picture
2. Put your `print('hello world')` code in it
3. Run your file using the F5 key

Adding a comment!

Sometimes we want to write things in our file that the computer doesn't look at. We can use **comments** for that!

Sometimes we want to write a note for a people to read

```
# This code was written by Vivian
```

And sometimes we want to not run some code (but don't want to delete it!)

```
# print("Goodbye world!")
```

Try it!

1. Add a comment to your hello.py file
2. Run your code to make sure it doesn't do anything extra!

Project time!

You now know all about printing and variables!

Let's put what we learnt into our project
Try to do the next Part!

The tutors will be around to help!