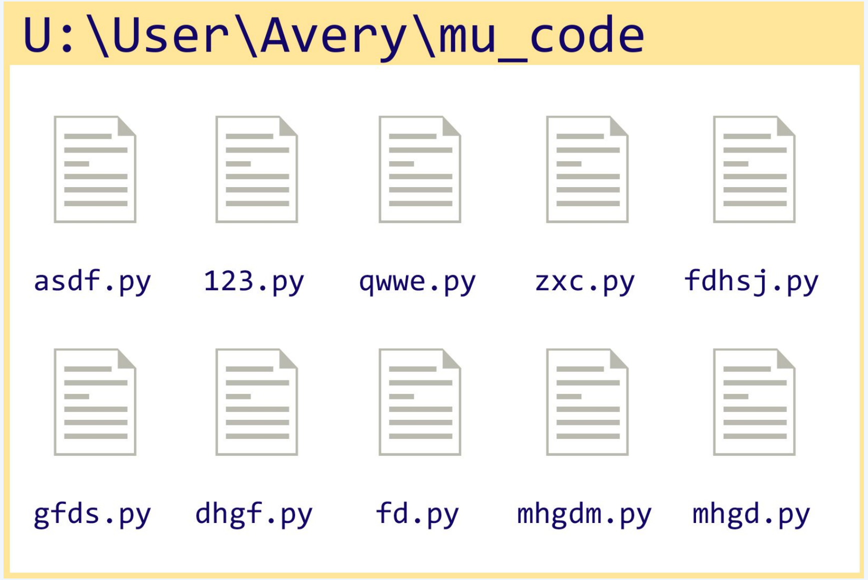
Lesson 2: Variables

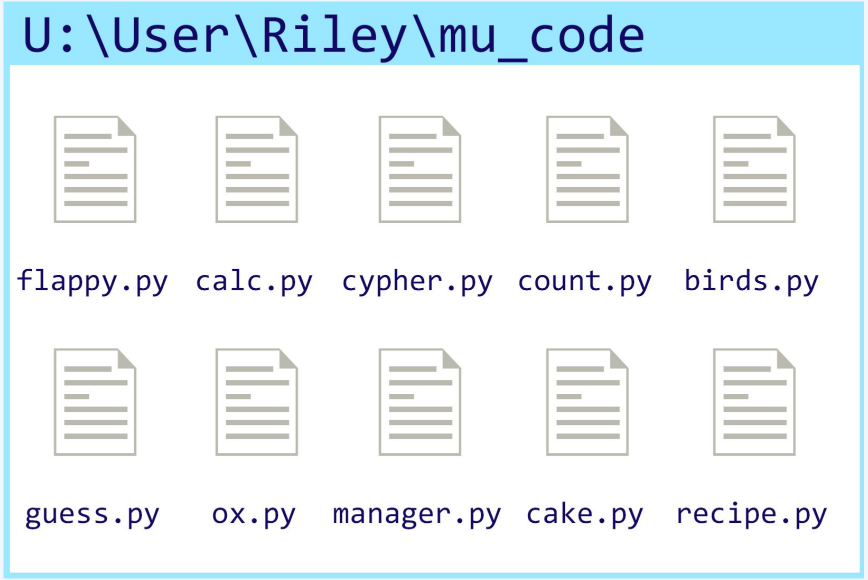
In pairs, compare and contrast these user areas



U:\User\Avery\mu_code

asdf.py 123.py qwwe.py zxc.py fdhsj.py

gfds.py dhgf.py fd.py mhgdm.py mhgd.py



U:\User\Riley\mu_code

flappy.py calc.py cypher.py count.py birds.py

guess.py ox.py manager.py cake.py recipe.py

Who has the better user area and **why?**

Why is it important to name files correctly?

Python naming conventions

In programming, there are **naming conventions** that should be followed to make it easier to read and understand your code.

Python has an agreed set of **conventions** that programmers should follow.

You will explore some of these during this lesson.

```
age

first_name

SPEED
```

Lesson 4: Variables

In this lesson, you will:

- Use meaningful identifiers
- Determine the need for variables
- Distinguish between declaration, initialisation, and assignment of variables
- Demonstrate appropriate use of naming conventions
- Output data (e.g. `print (my_var)`)

What can you do in Python?

Previously you have:

- Written a program in a **sequence**
- Used the **print** function
- Used a **subroutine**
- Checked for **errors** in your code
- **Fixed** common errors in your code

This lesson, you will be introduced to a new programming construct known as a **variable**.

What is a variable?

A **variable** holds a **value** in a **memory location** that is needed for the **execution** of your program.

A variable can hold **one value** at a time. This value can **change** throughout the execution of the program.

What is a variable?

You might use a **variable** to keep track of the **score** for a game.

At the beginning of a game the **score** might be 0.

score $\boxed{0}$

What is a variable?

You might use a **variable** to keep track of the **score** for a game.

At the beginning of a game the **score** might be 0.

During gameplay, the player might earn a point.

```
score       0
```

```
score = score + 1
```

What is a variable?

You might use a **variable** to keep track of the **score** for a game.

At the beginning of a game the **score** might be 0.

During gameplay, the player might earn a point.

This would change the value held in score by increasing the value by 1.

```
score        1
```

```
score = score + 1
```

Using a variable

A **memory location** needs to be allocated for the **variable value**, before it can be used by the program.

Declaring a variable means stating what **data type** will be used for the value.

Initialising a variable means setting the initial **value**.

Variable declaration in other programming languages

Python doesn’t use variable declaration, only initialisation.

Here are examples of **variable declaration** in **Visual Basic** and **C.**

Integer or **int** just means **whole numbers**.

Both of these statements mean **declare the variable age as an integer**.

Visual Basic

```
Dim age As Integer
```

C

```
int age;
```

Variable initialisation

Initialisation is when the **initial value** is **assigned** to the variable.

Remember that variables only hold **one** value at a time and this value can **change** throughout the **execution** of the program.

Here are examples of variable initialisation. This means **hold the value 22 under the variable name, age.**

Visual Basic

```
age = 22
```

C

```
age = 22;
```

Python

```
age = 22
```

Creating and using our first variable

Line 1 **initialises** the variable by calling it **name** and **assigning** the value **“Gerry”** at a memory location.

```
1   name="Gerry"
2   print(name)
3
```

Activity

1. Type this code into Python and run it to see what happens.
2. Swap lines 1 and 2 around and read the error message that occurs.

Creating and using our first variable

When you try to **use** a variable before it has been **initialised,** it will cause an error.

Instructions are executed one after the other.

In this case the error would be:

```
name, ‘name’ is not defined
```

```
print(name)
name="Gerry"

```

```
Traceback (most recent call last):
  File "c:\users\rebecca\mu_code\name.py",
line 1, in <module>
    print(name)
NameError: name 'name' is not defined
```

Creating and using our first variable

Once a variable has been **initialised** with its first value, it can then be **reassigned** a new value throughout the execution of the code.

```
name="Gerry"
name="Sam"
print(name)
```

Question

- What do you think might happen when this code is executed?

Creating and using our first variable

The program will display **Sam** on the screen. This is because it was the last value that was assigned to **name**.

```python
name="Gerry"
name="Sam"
print(name)
```

```
Sam
>>>
```

Meaningful identifiers and naming conventions

Variables should always be identified in a **meaningful way**, like the files from the starter activity.

a, b, c, x, y are not helpful names for a variable.

If a variable is going to hold a name, then call that variable **name**.

```
x = "Gerry"
```
✗

```
name = "Gerry"
```
✓

Meaningful identifiers and naming conventions

All programming languages have their own **naming conventions**.

A considerate programmer will read the documentation for the programming language to find out what the conventions are.

In Python, variable names should be written in **lower case,** with an **underscore** separating words if required.

ncce.io/pythondoc

Recap

Declaration means to declare a variable as a certain **data type**.

Initialisation means to set an **initial** value for a variable.

Assignment means to **change** the value held at the variable location.

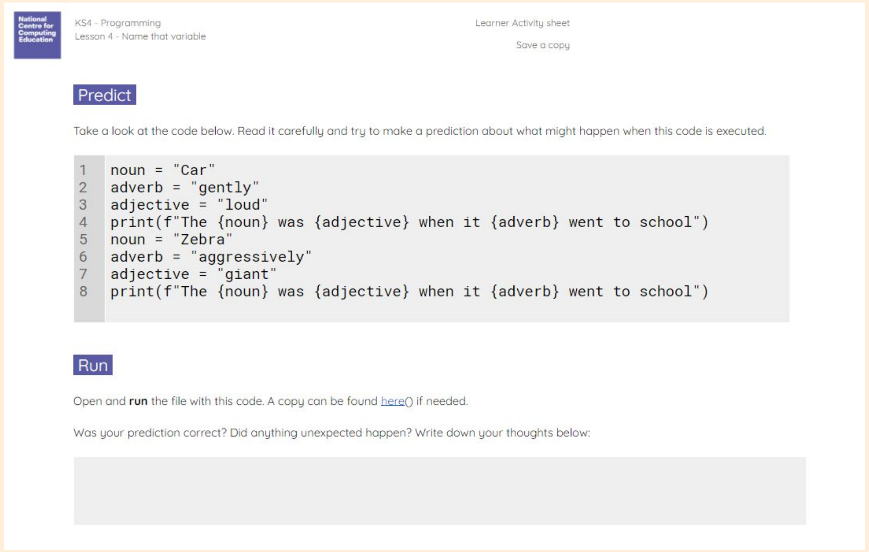
A **variable** must be **initialised** before it can be used.

Meaningful identifiers are essential.

Silly sentences

Use the **Activity 2 worksheet** to predict, run, investigate, and modify a silly sentences program.

The final task is to create your own silly story.



KS4 - Programming
Lesson 4 - Name that variable

Learner Activity sheet

Save a copy

Predict

Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```
noun = "Car"
adverb = "gently"
adjective = "loud"
print(f"The {noun} was {adjective} when it {adverb} went to school")
noun = "Zebra"
adverb = "aggressively"
adjective = "giant"
print(f"The {noun} was {adjective} when it {adverb} went to school")
```

Run

Open and **run** the file with this code. A copy can be found here() if needed.

Was your prediction correct? Did anything unexpected happen? Write down your thoughts below:

Peer-review your silly stories

Sit with a partner and peer-review your silly stories. A checklist has been provided to focus your review.

- ❑ Has the programmer used a variety of variables to fill in the blanks?
- ❑ Are there any syntax errors?
- ❑ Are they any logic errors?
- ❑ How could the story be improved?

Improve your silly stories

Based on the feedback from your partner, improve your silly stories program.

Due next lesson

Next lesson

In this lesson, you…

Learnt how to use variables in your programs.

Next lesson, you will…

Learn how to accept user input into your program.