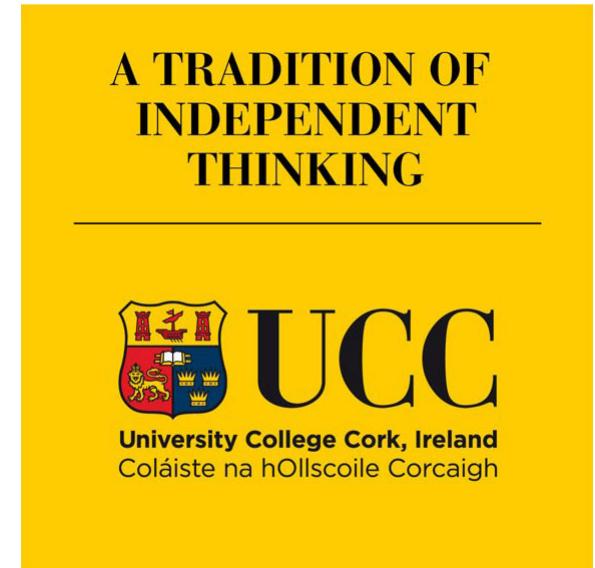




CS1117 – Introduction to Programming

Dr. Jason Quinlan,
School of Computer Science and Information Technology



Announcements

CANVAS

Can you see **CS1117** in the list of courses?

Are you auto-connecting to **CS1117**?

Announcements

EDUROAM (free college WiFi)

<https://www.ucc.ie/en/sit/wireless/>

To connect to the **eduroam** Wi-Fi service
simply select “**eduroam**” from the available Wi-Fi networks
and choose connect.

Your username **StudentNum@umail.ucc.ie** (e.g.
123456789@umail.ucc.ie) and
your password is your **Student IT Account** password.

Announcements

Labs & Assignments

If I say there is **no labs** this week,
It is for **this** module (CS1117)

Same for assignments, exams, etc.

Announcements

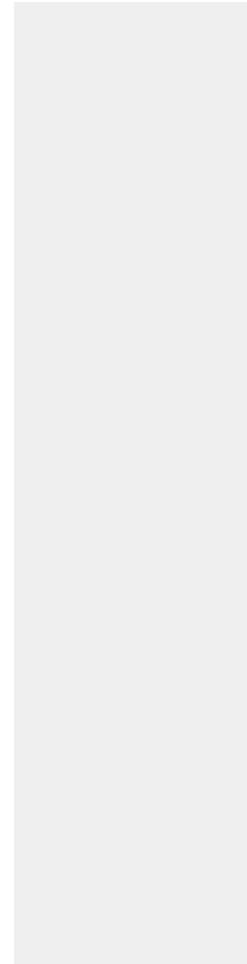


IDE install issues:

If anyone has installed Atom or PyCharm on their home computer or laptop and are having issues.

Let me know and I'll cover install in the next lecture.

If not, we will cover install completely in the labs next Tuesday and Wednesday.



Announcements

CS1117 Labs start next week:

Labs

Tues (17th Sept.) 4-6 G.20

Wed (18th Sept.) 4-6 G.20

Student Allocation for the Labs
will be posted to Canvas tomorrow

Announcements

Volunteer Research Assistant

Announcements

Volunteer Research Assistant

MISL Summer of Code is a personal initiative in which students from local secondary schools and third level institutions (foreign and domestic) intern in the Computer Science department of UCC, under my supervision.

Announcements

Volunteer Research Assistant

MISL Summer of Code is a personal initiative in which students from local secondary schools and third level institutions (foreign and domestic) intern in the Computer Science department of UCC, under my supervision.

Two interns attended in the summer of 2018

Five interns attended in the summer of 2019

Announcements

Volunteer Research Assistant

MISL Summer of Code offers some research

Lots of problem solving

Lots of coding, in a multitude of languages

Lots of fun!!

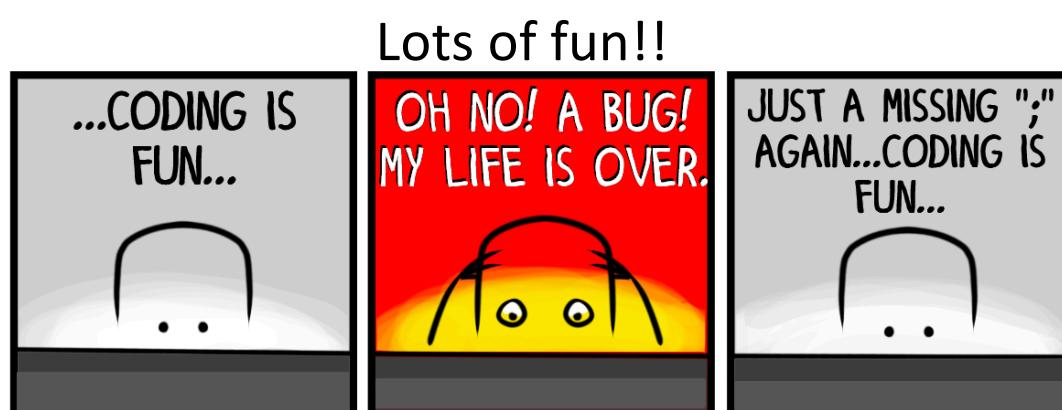
Announcements

Volunteer Research Assistant

MISL Summer of Code offers some research

Lots of problem solving

Lots of coding, in a multitude of languages



Announcements

Volunteer Research Assistant

Would you like to assist in research in the Mobile and Systems Laboratory (MISL) for a period of 8 weeks this semester?

Announcements

Volunteer Research Assistant

Would you like to assist in research in the Mobile and Systems Laboratory (MISL) for a period of 8 weeks this semester?

You will assist on projects an average of 5-10 hours per week.

Announcements

Volunteer Research Assistant

Would you like to assist in research in the Mobile and Systems Laboratory (MISL) for a period of 8 weeks this semester?

You will assist on projects an average of 5-10 hours per week.

You get your own desk.

Announcements

Volunteer Research Assistant

Would you like to assist in research in the Mobile and Systems Laboratory (MISL) for a period of 8 weeks this semester?

You will assist on projects an average of 5-10 hours per week.

You get your own desk.

Help develop *a unique set of skills*

Announcements

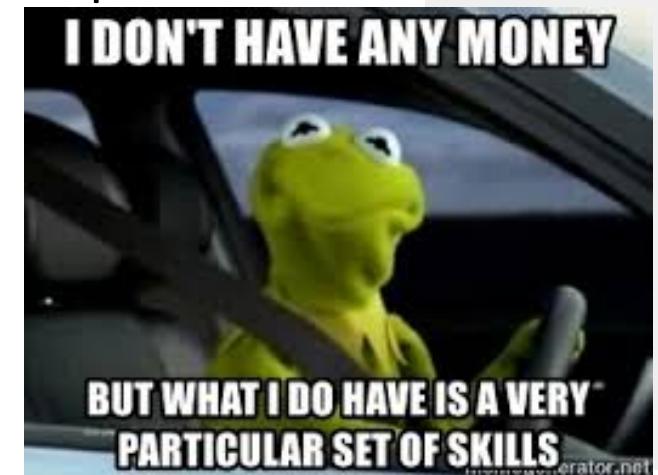
Volunteer Research Assistant

Would you like to assist in research in the Mobile and Systems Laboratory (MISL) for a period of 8 weeks this semester?

You will assist on projects an average of 5-10 hours per week.

You get your own desk.

Help develop *a unique set of skills*



Announcements

Volunteer Research Assistant

How do I apply for this amazing opportunity?

Email me: j.quinlan@cs.ucc.ie

Put “MISL VRA S1 19” in the subject line

That's it.

Final decision by others ☺

Announcements

Volunteer Research Assistant

Important:

If you are offered a volunteer research position, it is
voluntary, no payment, but more importantly:

It can not affect your other course work!!!

Any slip in the **quality** or **quantity** of your course work of all other modules will mandate your position being allocated to another student.

Announcements

Game Time!!!

Announcements

Game Time!!!

By the end of the first semester you will know how to implement the logic of this game.

Then we can have some fun in semester 2!!!

Announcements

Question Time!!!



► LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.



► LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.



► LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.



► LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.



► LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

QUICK RECAP

- Covered “*what is programming?*”
- Covered the 4 pillars of programming:
 - Problem, logic, code and test
- Introduced Pseudocode
- Asked a simple question about “*averaging ages*”
 - Wrote some pseudocode
 - Wrote some Python code based on the pseudocode
 - Went over the different elements of the Python code
 - Variables
 - Functions
 - Input from the keyboard
 - Casting (changing type) from int to string and vice versa
- Did some live coding...

QUICK RECAP

```
average_ages.py
1 # get the first age
2 age1 = int(input("Please enter age 1: "))
3 # get the second age
4 age2 = int(input("Please enter age 2: "))
5 # determine the average age
6 average = (age1+age2)/2
7 # print to screen
8 print("The average age is %d" % average)
```

This is our Python code

If you were not here, check out *Week 1* slides on Canvas

You will find them under the heading “**Modules**”

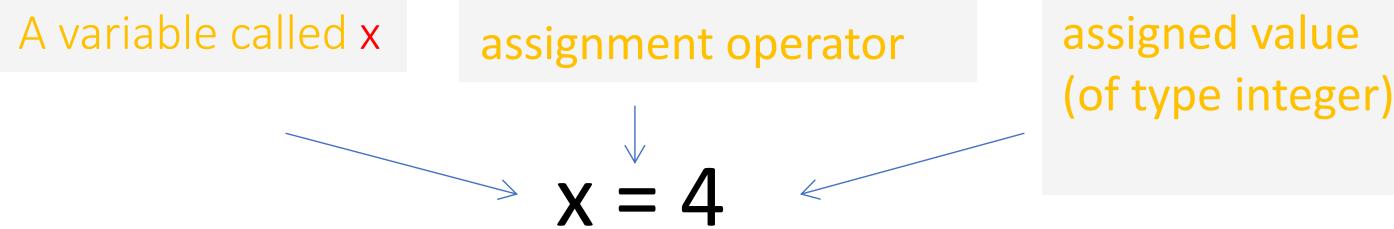
Announcements

Live Coding from Lecture 2

Quick review ☺

Data Types

- In lecture 2 code, we covered int, 4, and string, “4”



You usually *assign* (give) a value or piece of data
to a variable using the `=` operator

Data Types

Some of the other Data types available in Python

Type	Example
Numeric: Integer, Float	x = 10 x = 10.0
String	x = "Mike"
Boolean	x = True x = False
List	x = [10, 20, 30]
Tuple	x = ("Ed", "Edd", "Eddy", 2009)
Dictionary	x = {'one': 1, 'two': 2}
List	x = ["Ed", "Edd", "Eddy", 2009]

Data Types

If we write “`print(type(x))`” for each example, we get the type:

Type	Example	<code>type()</code>
Numeric: Integer, Float	<code>x = 10</code> <code>x = 10.0</code>	<code><class 'int'></code> <code><class 'float'></code>
String	<code>x = "Mike"</code>	<code><class 'str'></code>
Boolean	<code>x = True, x = False</code>	<code><class 'bool'></code>
List	<code>x = [10, 20, 30]</code>	<code><class 'list'></code>
Tuple	<code>x = ("Ed", "Edd", "Eddy", 2009)</code>	<code><class 'tuple'></code>
Dictionary	<code>x = {'one': 1, 'two': 2}</code>	<code><class 'dict'></code>
List	<code>x = ["Ed", "Edd", "Eddy", 2009]</code>	<code><class 'list'></code>

Data Types

```
print("x is a ", type(x), "and has a value of", x)
```

```
# data types
x = 10
print("x is a ", type(x), "and has a value of", x)
x = 10.0
print("x is a ", type(x), "and has a value of", x)
x = "Mike"
print("x is a ", type(x), "and has a value of", x)
x = True
x = False
print("x is a ", type(x), "and has a value of", x)
x = [10, 20, 30]
print("x is a ", type(x), "and has a value of", x)
x = ("Ed", "Edd", "Eddy", 2009)
print("x is a ", type(x), "and has a value of", x)
x = {'one': 1, 'two': 2}
print("x is a ", type(x), "and has a value of", x)
x = ["Ed", "Edd", "Eddy", 2009]
print("x is a ", type(x), "and has a value of", x)
```

Data Types

```
print("x is a ", type(x), "and has a value of", x)
```

```
x is a <class 'int'> and has a value of 10
x is a <class 'float'> and has a value of 10.0
x is a <class 'str'> and has a value of Mike
x is a <class 'bool'> and has a value of False
x is a <class 'list'> and has a value of [10, 20, 30]
x is a <class 'tuple'> and has a value of ('Ed', 'Edd', 'Eddy', 2009)
x is a <class 'dict'> and has a value of {'one': 1, 'two': 2}
x is a <class 'list'> and has a value of ['Ed', 'Edd', 'Eddy', 2009]
```

print("x is a ", type(x), "and has a value of", x)

Variables and Function Names

- We've talked a bit about using a sensible name for your variables and functions
- Something you and other coders can understand (without reading the comments)
- So my use of `x = 4` or `x = [10, 20, 30]` is not very descriptive...
- For example, let's assume `x` is a variable that holds the number of the class attended in a day

Variables and Function Names

- So `class = 4` is more descriptive
- Another example would be for `x = [10, 20, 30]`
- If we want to save an X, Y, Z coordinate: we might want to store the global location
- `global = [10, 20, 30]`
- Both `class` and `global` are descriptive variable names **but....**

Variables and Function Names

Write and run the code:

```
class = 4
|
global = [10, 20, 30]
```

```
Jasons-MacBook-Pro:code_snippets jasonquinlan$ python3 ./lecture_3.py
  File "./lecture_3.py", line 40
    class = 4
      ^
SyntaxError: invalid syntax
Jasons-MacBook-Pro:code_snippets jasonquinlan$ python3 ./lecture_3.py
  File "./lecture_3.py", line 43
    global = [10, 20, 30]
      ^
SyntaxError: invalid syntax
```

Python reserved keywords

Just known that within Python
these **33 reserved** keywords exist

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

Variables and Function Names

Change the variables names

```
class_num = 4
global_location = [10, 20, 30]

print("The", class_num, "th class is at the x, y, z location", global_location)
```

```
The 4 th class is at the x, y, z location [10, 20, 30]
```

Variables and Function Names

- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.

Variables and Function Names

- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.
- Remember – different version of the same word are different variables
- Age, age, AGE,

Variables and Function Names

- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.
- Remember – different version of the same word are different variables
- Age, age, AGE,
- So, any one tell me what these variable names are:
- “III” or “All” or “II”

Variables and Function Names

- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.
- Remember – different version of the same word are different variables
- Age, age, AGE,
- So, any one tell me what these variable names are:
- “**III**” or “**All**” or “**II**”
- “**ill**” (some one sick)

Variables and Function Names

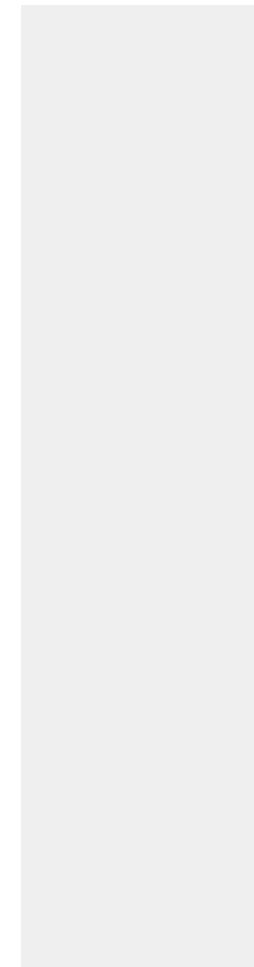
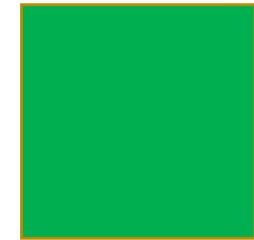
- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.
- Remember – different version of the same word are different variables
- Age, age, AGE,
- So, any one tell me what these variable names are:
- “**III**” or “**All**” or “**II**”
- “ill” (some one sick) or “ail” (to be ill)

Variables and Function Names

- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
- In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.
- Remember – different version of the same word are different variables
- Age, age, AGE,
- So, any one tell me what these variable names are:
- “III” or “All” or “II”
- “ill” (some one sick) or “ail” (to be ill) or “Li” (the symbol for Lithium)

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - `numberofcollegegraduates = 2500`



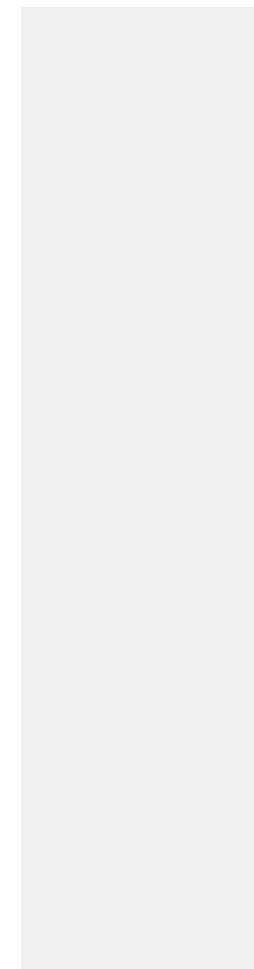
Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - `numberofcollegegraduates = 2500`

Lower Case: All words are lower case

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - `numberofcollegegraduates = 2500`
 - `NUMBEROFCOLLEGEGRADUATES = 2500`



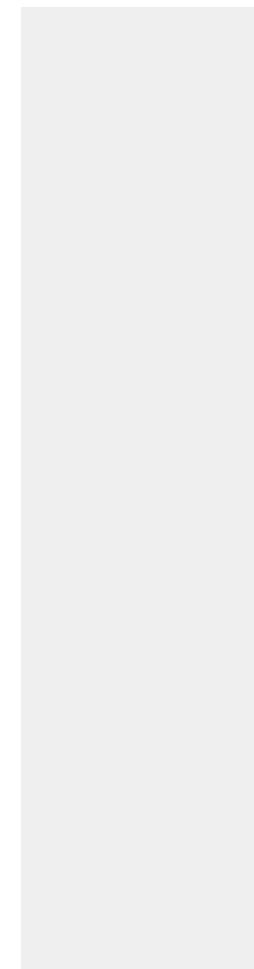
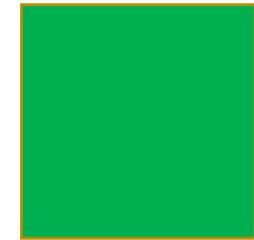
Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500

Upper Case: All words are upper case

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500



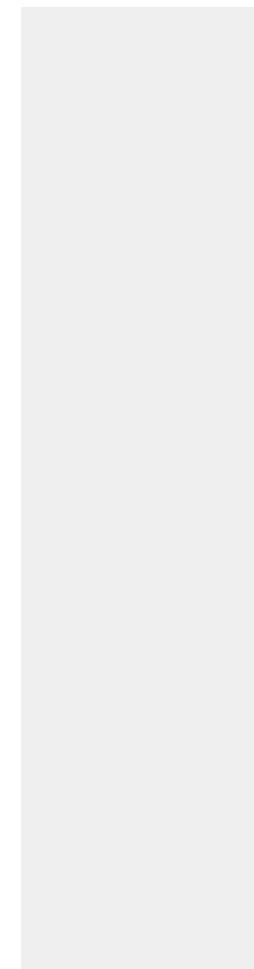
Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500

Camel Case: Second and subsequent words are capitalized, to make word boundaries easier to see.

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500



Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500

Pascal Case: Identical to Camel Case, except the first word is also capitalized.

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500
 - number_of_college_graduates = 2500

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500
 - number_of_college_graduates = 2500

Snake Case: All words are lower case

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500
 - number_of_college_graduates = 2500
- The [Style Guide for Python Code](#), also known as **PEP 8**
- <https://www.python.org/dev/peps/pep-0008/#names-to-avoid>
- “Variable/Function names should be lowercase, with words separated by underscores as necessary to improve readability.” – **Snake Case**

Variables and Function Names

- Python has some conventions (rules) for writing names:
- We could write:
 - numberofcollegegraduates = 2500
 - NUMBEROFCOLLEGEGRADUATES = 2500
 - numberOfCollegeGraduates = 2500
 - NumberOfCollegeGraduates = 2500
 - number_of_college_graduates = 2500
- The [Style Guide for Python Code](#), also known as **PEP 8**
- <https://www.python.org/dev/peps/pep-0008/#names-to-avoid>
- “Variable/Function names should be lowercase, with words separated by underscores as necessary to improve readability.” – **Snake Case**
- “Class names should normally use the CapWords convention” – **Pascal Case**

Invalid Variable Names

- Finally, Python has a naming convention for what can not be in the variable/function name:
- The name can not start with a number
 - 7_days_a_week

Invalid Variable Names

- Finally, Python has a naming convention for what can not be in the variable/function name:
- The name can not start with a number
 - 7_days_a_week
 - 1e1
 - Could this be 10.0 (maths notation) or a variable??

Invalid Variable Names

- Finally, Python has a naming convention for what can not be in the variable/function name:
- The name can not start with a number
 - 7_days_a_week
 - 1e1
 - Could this be 10.0 (maths notation) or a variable??
- The name can not have the “#” anywhere in it
 - #_of_days_a_week
 - Mixture#3

Invalid Variable Names

- Finally, Python has a naming convention for what can not be in the variable/function name:
- The name can not start with a number
 - 7_days_a_week
 - 1e1
 - Could this be 10.0 (maths notation) or a variable??
- The name can not have the “#” anywhere in it
 - #_of_days_a_week
 - Mixture#3

Why??

Invalid Variable Names

- Finally, Python has a naming convention for what can not be in the variable/function name:
- The name can not start with a number
 - 7_days_a_week
 - 1e1
 - Could this be 10.0 (maths notation) or a variable??
- The name can not have the “#” anywhere in it
 - #_of_days_a_week
 - Mixture#3



A screenshot of a dark-themed code editor showing two examples of invalid variable names. The first line contains the text "mixture#3 = 2", where the "#" character is highlighted in red. The second line contains the text "1e1 = 7", where the "1" is highlighted in orange and the "e1" is highlighted in purple, indicating multiple errors.

Variables Recap

- How we assign a variable a value
- Data types
- Variable naming
 - Reserved words
 - Ambiguity in naming - I (i) and I similarity
 - Camel and Pascal case naming
 - Do not use a number as first character (7_weeks)
 - Do not use “#” in the variable name

And this is only Python variables 😊



WUBBA LUBBA DUB DUB!



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh