

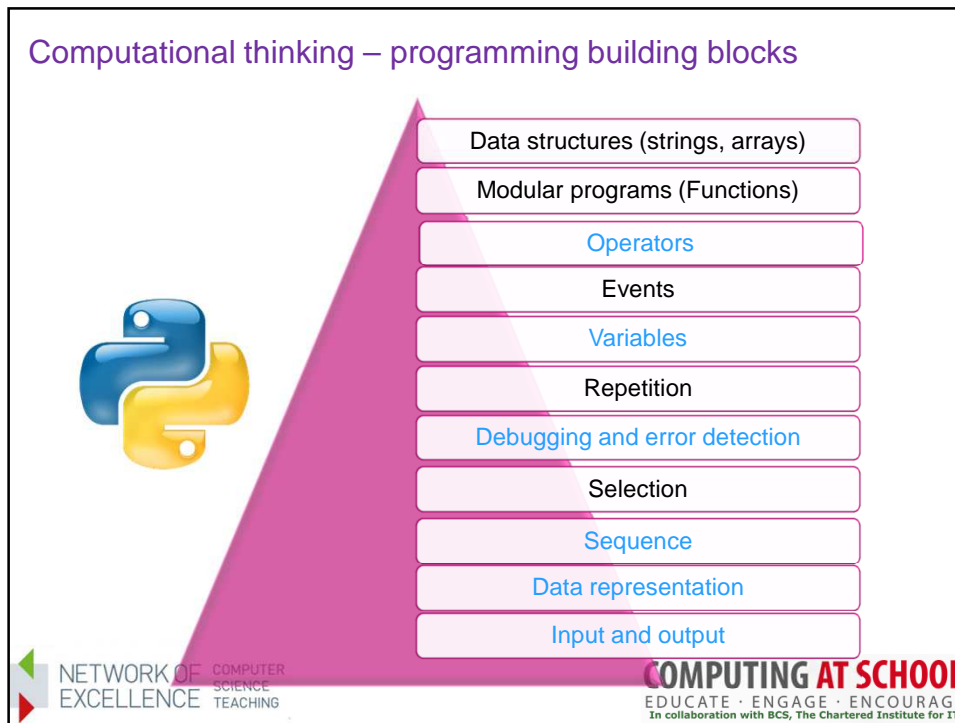
Python Lesson 2 If command and conditions

Suffolk One, Ipswich, 4:30 to 6:00
Tuesday Jan 21
Nicky Hughes



All programs can be represented
in terms of sequence, selection
and iteration.






Comparison operators and if command

Learning outcomes

- Review last week lessons
- Be able to use comparison operators (conditions)
- Be able to use the if else and elif commands
- Be able to use string formatting
- Write programs to solve problems



NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Which part is which?

```
# This program displays the food order
numberGuests = 90
FoodOne = "pizza"
print("Please order", numberGuests, FoodOne)
```

Label the variable assignments.

Label the strings.

Label the function.

Label the arguments to the function

Label the comment.

What colour coding does IDLE use?

Program construct	Colour code
string	
variable	
function	
comment	

Name the four main data types that Python uses.

<u>Name of data type</u>	<u>Python abbreviation</u>	<u>Example of the data type</u>
	int	
	str	
	float	
	bool	

What are these data types?

Expression	Data type	Try this command
"hello world"		type("hello world")
False		type(False)
15		type(15)
35.6		type(35.6)
-999		type(-999)
"15"		type("15")
"False"		type("False")
True		type(True)
0.001		type(0.001)

Look at this program.
What does the program do?

```
age=input("Please enter your age: ")
agePlusTen = age + 10
print("You will be", agePlusTen, "in 10 years")
```

Enter the program into a script and run the program.

What happens? Interpret the error message.

Amend the program so that it runs successfully.

Are these valid or invalid variable names?
Explain your answer.

	Valid or invalid?	Reason
8HouseNumber = 288		
houseNumber = 288		
house Number = 288		
house_number = 288		
import = 288		

What is the correct answer to the following expression? Explain your answer.

$$>>> 7 + 4 * 5$$

- A.55
- B.27
- C.16
- D.33

What is the correct answer to the following expression? Explain your answer.

>>> 6 - 2 / 2 + 5

- A. 0.57
- B. 10
- C. 7
- D. 11

Conditional if else command

Condition statements decide which commands to run in a program.

Condition statements are either **True** or **False**.

If the condition is "True" one sequence of commands is run and if the condition is "False" another set of commands is run.

```
if <condition is true> :
    command x
else :
    command y
```

Comparison operators allow us to compare things. The answer is always True or False

The condition tests values using the **comparison operators**.

<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code><</code>	Less than
<code>></code>	Greater than
<code><=</code>	Less than or equal to
<code>>=</code>	Greater than or equal to

Use the interactive shell to see how comparison return True or False

```

>>> 5 < 8
True
>>> 10 < 8
False
>>> colour = "red"
>>> colour == "green"
False
>>> colour == "red"
True
>>> 99 <= 99
True
>>>
  
```

The shell returns whether the boolean value "True" or "False"

Are the following answers True or False?

```
>>> "Friday" == "Sunday"
```

```
>>> 45 < 7
```

```
>>> 9 > 1
```

```
>>> 5 <= 5
```

```
>>> 8 != 8
```

```
>>> 101 == 102
```

Are the following answers True or False?
Using variables to represent numbers

```
>>> numberOne = 9
```

```
>>> numberTwo = 8
```

```
>>> numberOne == numberTwo
```

```
>>> numberOne < numberTwo
```

```
>>> numberOne >= numberTwo
```


Are the following answers True or False?
 Using variables to represent strings.
 Strings are compared alphabetically using the
 ASCII numeric equivalents of their characters

```
>>> stringA = "red"
```

```
>>> stringB = "green"
```

```
>>> stringA == stringB
```

```
>>> stringA != stringB
```

```
>>> stringA < stringB
```



ASCII BINARY CHARACTER TABLE (extract)

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000



<http://sticksandstones.kstrom.com/appen.html>
ASCII to binary table

The ord() function converts from letters to numbers in Python. It returns the decimal ASCII number that represents that character.

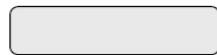
```
>>> ord("a")
97
>>> ord("b")
98
```

FLOW CHART SYMBOLS (revision)

Flowchart symbols



Denotes the beginning and end of an algorithm.



Denotes a process to be carried out.



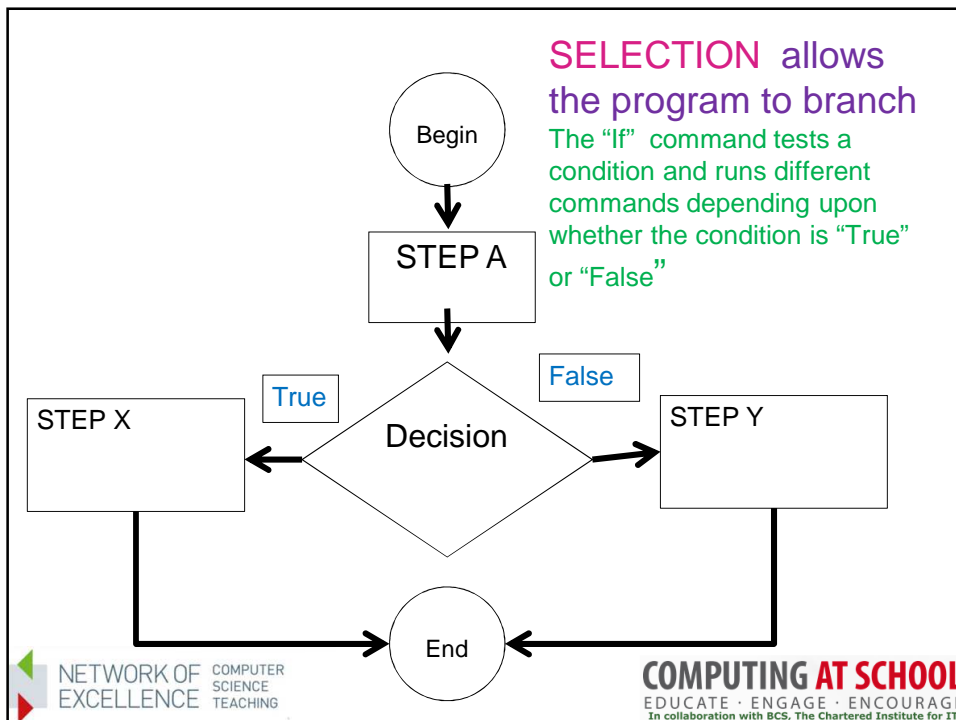
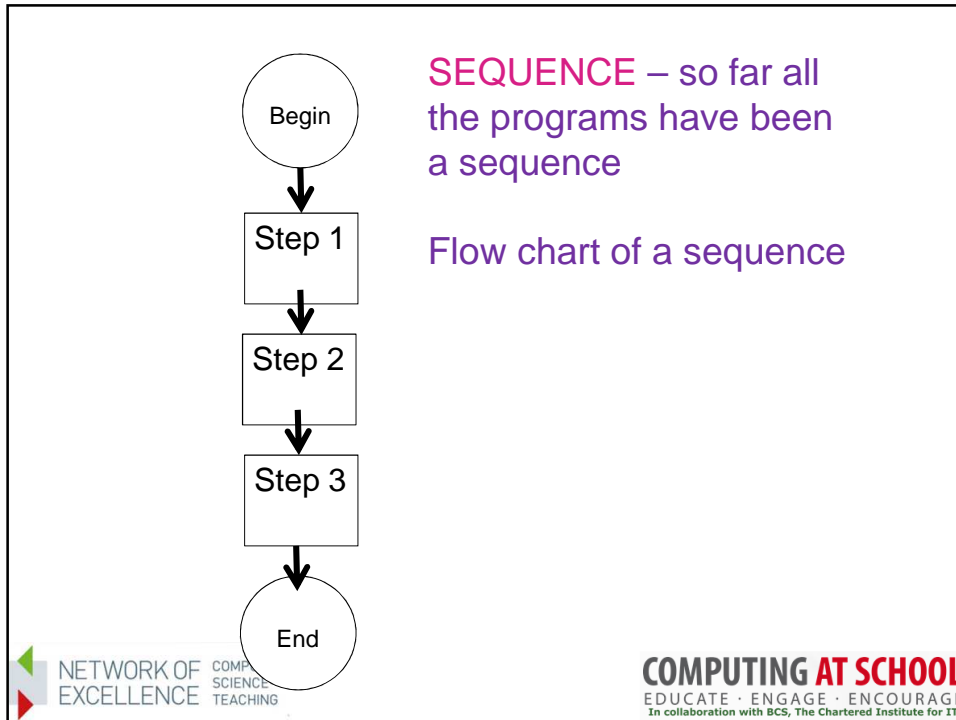
Denotes a decision to be made.



Shows the logical flow of the program.

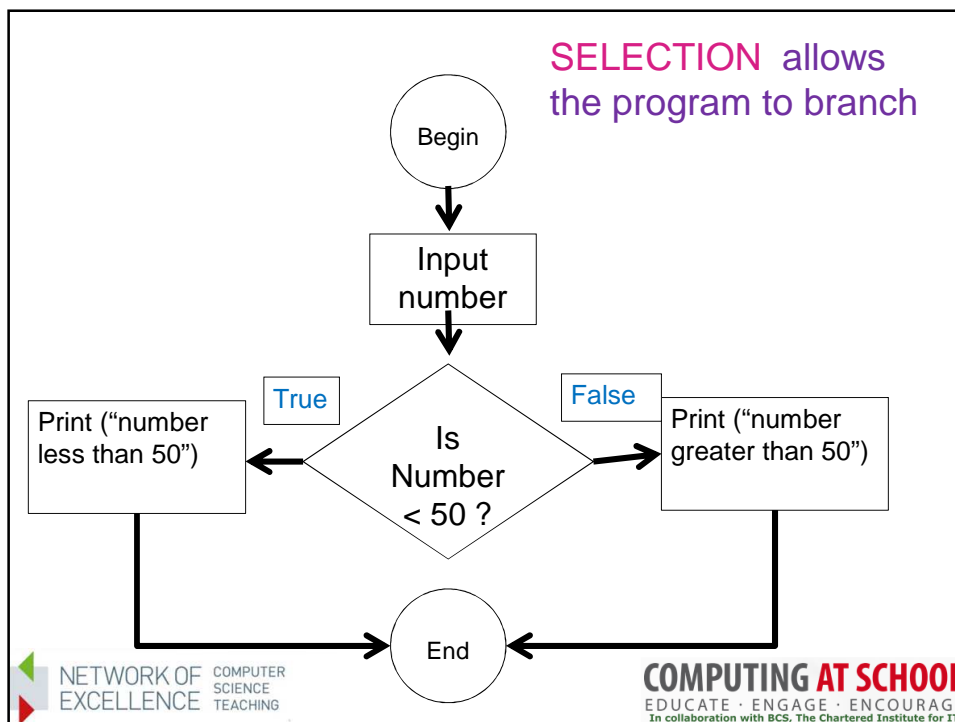
Show the YouTube video 'The Big Bang Theory – The Friendship Algorithm':

<http://www.youtube.com/watch?v=k0xjUjHEG3U>



Challenge:

Input a number and display whether the number is less than or greater than 50.



If commands

Syntax	Description	Example
if <expression>: <commands>	<i>If <expression> is true then commands are executed.</i>	if answer=="Y": print("You may continue")
if <expression>: <commands1> else: <commands2>	<i>If <expression> is true then <commands1> are executed otherwise <commands2> are executed</i>	if colour == "green": print("It is safe to cross.") else: print("STOP!")

```

number = int(input("Please enter a number"))
if number < 50 :
    print("Number is less than 50")
else:
    print("Number is greater than 50")
  
```

If and else are compound statements which end with a colon (orange)

Indentation indicates start and end of the commands for each logical path.

Challenge

Write a program that asks if it is raining.
If the answer is "Y" then display "Please take an umbrella" else display "Have a good day".

```

answer = input("Is it raining? Y or N ")
if answer == "Y":
    print("Take an umbrella")
else:
    print("Have a good day!")
  
```

THINGS TO CHECK if your program does not work

- Have you used == in the condition?
- Have you used = to assign a value to a variable
- Is there a colon at the end of "if" and the "else"?
- Are the commands indented by 5 spaces under "if" and "else"
- Is spelling of the variable names the same throughout?

Try the challenges

Challenge C1

The correct password to enter a computer system is "friend". Write a program that asks the user to input a password.

If the "password == "friend" " is true the program displays "Access granted".

If it is false the program displays "Access denied".

```
password = input("Enter password: ")
if password == "friend":
    print("Access granted")
else:
    print("Access denied")
```

Enter the program code and test the program works with both valid and invalid passwords.
Change the program to accept a different password.

elif (like nested if else or a CASE statement)

Syntax	Description
if <expressionA>: <commands1> elif <expressionB>: <commands2> elif <expressionC>: <commands3> else: <commands4>	<i>If <expressionA> is true then <commands1>are executed, else if <expressionB> is true then <commands2>are executed, etc else <commands4 are executed.</i>

```

mark = int(input("Enter your mark"))
if mark >= 80:
    result = "A"
elif mark >= 60:
    result = "B"
elif mark >= 40:
    result = "C"
else :
    result = "D"
print("your result is", result) |

```

elif command Challenge 7

- This program asks the user to enter a mark and then gives a grade for that mark. Run the program a few times entering different marks between 0 and 100.
- What range of marks gets each grade?

Combining conditions using AND & OR

Conditions can be combined using the keywords and & or

<condition> **or** <condition> will be True if either conditions is true

<condition> **and** <condition> will be True if both conditions is true

String formatting

The string method `.format` gives you more control over the formatting of output.

Use format fields (placeholders) for variables within strings.

The numbers in the `{ }` brackets are replaced by the objects given in the `.format` method. They refer to the position of the objects and start from zero.

String formatting is like a mini-language all of its own.



Try these string formatting examples: What will they do?

```
>>> foodOne = "fish"
>>> foodTwo = "chips"
>>> print("{0} and {1}".format(foodOne, foodTwo))

>>> print("{1} and {0}".format(foodOne, foodTwo))

>>> print("{1}{1}{1} and {0}{0}{0}".format(foodOne, foodTwo))|
```



- Use string formatting to display numbers

```
>>>number = 765.87641987
```

```
>>>print("The answer is {0:.5f}".format(number))
```

- green is the print command
- yellow is the string
- pink is the field format (place holder). 0 is the first object in the format . *The ‘:’ is a delimiter, the ‘.5’ gives the number of decimal places and the ‘f’ means fixed point.*
- grey is the .format method
- blue is the name of the variable. As it is the first variable it is referred to as “0” in the field format.

- Use string formatting to display numbers

```
>>>number = 765.87641987
```

```
>>>print("The answer is {0:.5f}".format(number))
```

Display the number to 6, 4, 2 and 0 decimal places.

```

>>> print("The number is {0:.6f}".format(number))
The number is 765.876420
>>> print("The number is {0:.4f}".format(number))
The number is 765.8764
>>> print("The number is {0:.2f}".format(number))
The number is 765.88
>>> print("The number is {0:.0f}".format(number))
The number is 766
>>> print(number)
765.87641987

```

Assign the number 2150.728932938 to a variable.

Use string formatting to display the number to 6,4,2 and 0 decimal places.

Useful resources

<http://www.pythonschool.net/>

<http://www.edexcel.com/quals/gcse/gcse-2013/computer-science/Pages/default.aspx>

Downloadable scheme of work to teach Python (Year 10)