if, while & Boolean algebra

There are many learning and practice resources for Python on the web, often fee to use. Examples are [www.snakify.org](http://www.snakify.org) , [www.w3schools.com](http://www.w3schools.com) , [www.freecodecamp.org](http://www.freecodecamp.org) and [www.realpython.com](http://www.realpython.com) . Please do make use of them

# Section – if

## Part 1

You can type ‘if’ statements directly into IDLE and execute them. Try this:

>>> if ‘three’ > ‘four’:

print(“ ‘three’ > ‘four’ “)

That’s great just for trying-things-out but usually you will want to save your work in a script, or better still, generalise the algorithm and wrap it into a reusable function definition:

Create and test these functions in a python file named w3starter.py

def my\_max\_v1(a,b):

if a > b:

return a

if b >= a

return b

def my\_max\_v2(a,b):

if a > b:

return a

return b

my-max = my\_max\_v2

Here are few more simple functions using ‘if’ to practice with:

def grade\_v1(mark):

""" Pass or Fail? """

if mark > 60:

return 'Pass'

return 'Fail'

def weather\_v1(a\_str):

""" take umbrella? """

if 'rain' in a\_str:

return 'Take umbrella'

def weather\_v2(a\_str):

""" rain or shine? """

if 'rain' in a\_str:

return 'Take umbrella'

if 'sun' in a\_str:

return 'Take sunscreen'

return ‘Look outside and guess’

def exec\_decider(deadline\_today=True):

if deadline\_today:

return 'Do it now!'

return 'mañana'

Run w3starter.py then, after checking the functions you wrote are now in memory by using the built-in function dir(), type this code into the IDLE interactive shell. Is the result what you expected?

>>>

>>> my\_max( 'three' > 'four')

>>> my\_max(3, 4)

Test your functions to make sure they work as anticipated.

## Part 2 – else

def grade\_v2(mark):

""" Pass or Fail? """

if mark > 60:

return 'Pass'

else:

return 'Fail'

## Part 3 - elif

def grade\_v3(mark):

if student\_mark > 80:

grade = ‘starship captain’

else:

if student\_mark > 80:

grade = ‘astronaut’

else:

if student\_mark > 60:

grade = ‘aeronaut’

else:

if student\_mark > 40:

grade = ‘earthling’

else:

grade = ‘oh deary, deary me’

def grade\_v4(mark):

if student\_mark > 80:

grade = ‘Starship Captain’

elif student\_mark > 80:

grade = ‘Astronaut’

elif student\_mark > 60:

grade = ‘Aeronaut’

elif student\_mark > 40:

grade = ‘Earthling’

else:

grade = ‘Oh deary, deary me’

## Section – Boolean Algebra

Evaluate these Boolean equations on paper, then confirm by executing them in IDLE:

.>>> a = True

>>> b = True

>>> a and b

>>> a or b

>>> not a and b

>>> not a or b

>>> not not not not a

>>> a and not a

>>> a or not a

## Section - Truth Tables and Venn Diagrams

These can be helpful when evaluating Boolean expressions in ‘if’ statements and elsewhere. Not assessed but very handy tools when you need them. For instance:

(a or not b) and (not b or not a) or (b and not a)

That said though, it is good practice to design your code so that Boolean conditions are individually simple.

## Section – while statements

Try these examples:

def counting(upto = 10):

counter = 0

while counter < upto:

print(counter)

counter = counter + 1

return

# end of indented code marks end of while loop

Print just the vowels and spaces in a string:

Def print\_vowels(a\_str):

a\_str = 'the quick brown fox jumps over the lazy dog'

vowels = 'aeiou ' # note the space at end of string

index = 0

while index < len(a\_str):

if a\_str[index] in vowels:

print(a\_str[index], end = ‘’)

else:

print(‘\_’, end = ‘’)

index = index + 1

Printing the characters of an alphabet:

def alphabet\_v1():

char\_code = ord('a')

stop = char\_code + 26

while char\_code <= stop:

print(chr(char\_code), end = ‘’)

char\_code = char\_code + 1

And other character sets:

def alphabet\_v2(start\_code, stop\_code):

“””Get start\_code & stop\_code from unicode.org/charts. Numbers are hexadecimal so precede number with 0x e.g. alphabet\_v2(0x00ff”””

counter = start\_code

while counter <= stop\_code:

print(chr(counter), end = ‘’)

counter = counter + 1

# Section – more practice

Go to [www.w3schools.com](http://www.w3schools.com) and work through the functions tutorial and tests <https://www.w3schools.com/python/python_functions.asp>

By now you should be familiar with most of the content within the following sections of W3Schools:

<https://www.w3schools.com/python/python_comments.asp>

<https://www.w3schools.com/python/python_variables.asp>

<https://www.w3schools.com/python/python_numbers.asp>

<https://www.w3schools.com/python/python_casting.asp>

<https://www.w3schools.com/python/python_strings.asp>

<https://www.w3schools.com/python/python_booleans.asp>

<https://www.w3schools.com/python/python_operators.asp>

<https://www.w3schools.com/python/python_functions.asp>