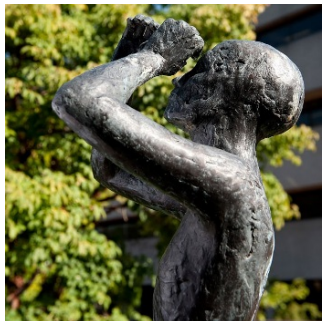


CS1117 – Introduction to Programming

Dr. Jason Quinlan,

School of Computer Science and Information Technology

A TRADITION OF INDEPENDENT THINKING

UCC

University College Cork, Ireland

Coláiste na hOllscoile Corcaigh

Announcements

Continuous Assessment 1

Wednesday – 23rd October 3-4pm in room 107

Multiple Choice Questions

Announcements

Continuous Assessment 2

Canvas access from 12th November 9am

Submission deadline 23rd November 1am

Covering Lectures from week 1 to week 9

No Christmas Exam ☺

While

One example for controlling our while loop using user input

```
print("Printing odd numbers")
limit = int(input("Provide a maximum number >>> "))

i = 1
while i < limit:
    print(i, end=" ")
    i += 2

print("\nPhew. The While has stopped")

# Output
# Provide a maximum number >>> 10
# 1 3 5 7 9
# Phew. The While has stopped
```

While

One example for controlling our while loop using user input

```
print("Printing odd numbers")
limit = int(input("Provide a maximum number >>> "))

i = 1
while i < limit:
    print(i, end=" ")
    i += 2

print("\nPhew. The While has stopped")

# Output
# Provide a maximum number >>> 10
# 1 3 5 7 9
# Phew. The While has stopped
```

While

One example for controlling our while loop using user input

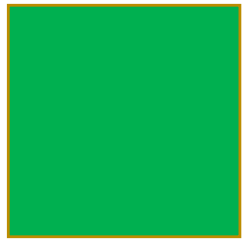
```
print("Printing odd numbers")
limit = int(input("Provide a maximum number >>> "))

i = 1
while i < limit:
    print(i, end=" ")
    i += 2

print("\nPhew. The While has stopped")

# Output
# Provide a maximum number >>> 10
# 1 3 5 7 9
# Phew. The While has stopped
```

While

One example for controlling our while loop using user input

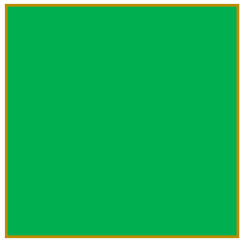
```
print("Printing odd numbers")
limit = int(input("Provide a maximum number >>> "))

i = 1
while i < limit:
    print(i, end=" ")
    i += 2

print("\nPhew. The While has stopped")

# Output
# Provide a maximum number >>> 10
# 1 3 5 7 9
# Phew. The While has stopped
```

While



Let’s see if we can write some code to convert a word/phrase into snake case…

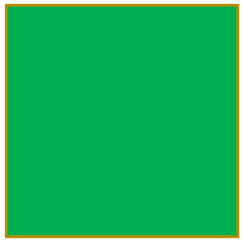
While



Let’s see if we can write some code to convert a word/phrase into snake case…

“hello world” => “h_e_l_l_o_w_o_r_l_d”

While

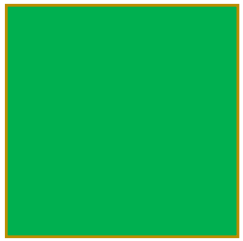


Let’s see if we can write some code to convert a word/phrase into snake case…

“hello world” => “h_e_l_l_o_w_o_r_l_d”

Let’s ask the user for a phase

While



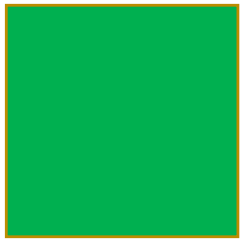
Let’s see if we can write some code to convert a word/phrase into snake case…

“hello world” => “h_e_l_l_o_w_o_r_l_d”

Let’s ask the user for a phase

And use a while loop for the conversion

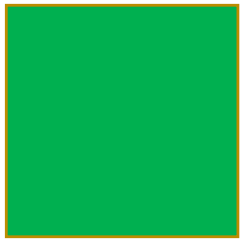
While



Snake case …

```
word = input("Please input a word/phrase >>> ")
```

While



Snake case …

```
word = input("Please input a word/phrase >>> ")
```

Let’s get some input from the user

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
```

Let’s get the length of the word/phrase, as we need this for the while loop

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
```

Let's set the initial value of our counter

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
```

Let's check the initial value of our counter against the length of our word

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
```

Let's check the initial value of our counter against the length of our word

If the condition is True

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
```

Let's print each character, using list indexing

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
```

And use end to place an underscore after the character

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
```

Let’s increment our counter

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
print()
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
print()
```

And print a return carriage at the end of the code

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_ _w_o_r_l_d_
```

While

Snake case …

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    print(word[i], end="_")
    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_ _w_o_r_l_d_
```

Opps…

While

Snake case … let's try slightly different code

Let's remove the spacing

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if word[i] != " ":
        print(word[i], end="_")
    i += 1
print()
```

While

Snake case … let's try slightly different code

Let's remove the spacing

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if word[i] != " ":
        print(word[i], end="_")
    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the spacing

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if word[i] != " ":
        print(word[i], end="_")
    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if word[i] != " ":
        print(word[i], end="_")
    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    if i == word_size-1:
        print(word[i])

    i += 1
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    if i == word_size-1:
        print(word[i])

    i += 1
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    if i == word_size-1:
        print(word[i])

    i += 1

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_d
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    if i == word_size-1:
        print(word[i])

    i += 1

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_d
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

Order matters!!!

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d_
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

Reverse the if and elif conditional statements

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:

    if word[i] != " ":
        print(word[i], end="_")

    elif i == word_size-1:
        print(word[i])

    i += 1
print()

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

Reverse the if and elif conditional statements

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if i == word_size-1:
        print(word[i])

    elif word[i] != " ":
        print(word[i], end="_")

    i += 1
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

We need to check for last letter before we check for not a space

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if i == word_size-1:
        print(word[i])

    elif word[i] != " ":
        print(word[i], end="_")

    i += 1
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

Reverse the if and elif conditional statements

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if i == word_size-1:
        print(word[i])

    elif word[i] != " ":
        print(word[i], end="_")

    i += 1

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d
```

While

Snake case … let's try slightly different code

Let's remove the trailing underscore – let's use if/elif

Reverse the if and elif conditional statements

```
word = input("Please input a word/phrase >>> ")
word_size = len(word)
i = 0
while i < word_size:
    if i == word_size-1:
        print(word[i])

    elif word[i] != " ":
        print(word[i], end="_")

    i += 1

# Output
# Please input a word/phrase >>> hello world
# h_e_l_l_o_w_o_r_l_d
```

While

Snake case … examples

```
Please input a word/phrase >>> hello world
h_e_l_l_o_ _w_o_r_l_d_
Please input a word/phrase >>> hello world
h_e_l_l_o_w_o_r_l_d_
Please input a word/phrase >>> hello world
h_e_l_l_o_w_o_r_l_d_d
Please input a word/phrase >>> hello world
h_e_l_l_o_w_o_r_l_d_
Please input a word/phrase >>> hello world
h_e_l_l_o_w_o_r_l_d
```

While

Snake case … examples

```
Please input a word/phrase >>> print()
p_r_i_n_t_(_)_
Please input a word/phrase >>> print()
p_r_i_n_t_(_)_
Please input a word/phrase >>> print()
p_r_i_n_t_(_)_)
Please input a word/phrase >>> print()
p_r_i_n_t_(_)_
Please input a word/phrase >>> print()
p_r_i_n_t_(_)
```

While

Snake case … examples

```
Please input a word/phrase >>> snake case
s_n_a_k_e_ _c_a_s_e_
Please input a word/phrase >>> snake case
s_n_a_k_e_c_a_s_e_
Please input a word/phrase >>> snake case
s_n_a_k_e_c_a_s_e_e
Please input a word/phrase >>> snake case
s_n_a_k_e_c_a_s_e_
Please input a word/phrase >>> snake case
s_n_a_k_e_c_a_s_e
```

While

Snake case … examples

```
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]_
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]_
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]_]
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]_
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]
```

While

Snake case … examples

```
word = list(input("Please input a word/phrase >>> "))
```

While

Snake case … examples

```
word = list(input("Please input a word/phrase >>> "))
```

```
Please input a word/phrase >>> ['s','n','a','k','e','c','a','s','e']
<class 'list'>
['[', "'", 's', "'", ',', "'", 'n', "'", ',', "'", 'a', "'", ',', "'", 'k', "'", ',', "'", 'e', "'", ',', "
'", 'c', "'", ',', "'", 'a', "'", ',', "'", 's', "'", ',', "'", 'e', "'", ']']
[_'_s_'_,_'_n_'_,_'_a_'_,_'_k_'_,_'_e_'_,_'_c_'_,_'_a_'_,_'_s_'_,_'_e_'_]
```

While

Snake case … examples

```
Please input a word/phrase >>> snakecase
<class 'list'>
['s', 'n', 'a', 'k', 'e', 'c', 'a', 's', 'e']
s_n_a_k_e_c_a_s_e
```

While

Few more examples:

```
# print values between 2 integers inclusive
def printValues(val1, val2):
    i = val1
    while i <= val2:
        print(i)
        i += 1


printValues(2, 5)

# Output
# 2
# 3
# 4
# 5
```

While

Few more examples:

```
# print values between 2 integers inclusive
# sum all the values together and print them out
def print_added_values(val1, val2):
    i = val1
    accum = 0
    while i <= val2:
        print(i)
        accum += i
        i += 1
    print("sum of all values:", accum)


print_added_values(2, 5)

# Output
# 2
# 3
# 4
# 5
# sum of all values: 14
```

While

Few more examples:

```
# print values between 2 integers inclusive
# sum all the values together and print them out
def print_added_values(val1, val2):
    i = val1
    accum = 0
    while i <= val2:
        print(i)
        accum += i
        i += 1
    print("sum of all values:", accum)


print_added_values(2, 5)

# Output
# 2
# 3
# 4
# 5
# sum of all values: 14
```

An accumulator allows us to store a value over the duration of the while loop

While

Few more examples:

```
# print values between 2 integers inclusive
# sum all the values together and print them out
def print_added_values(val1, val2):
    i = val1
    accum = 0
    while i <= val2:
        print(i)
        accum += i
        i += 1
    print("sum of all values:", accum)


print_added_values(2, 5)

# Output
# 2
# 3
# 4
# 5
# sum of all values: 14
```

An accumulator allows us to store a value over the duration of the while loop

While

Few more examples:

```
# print values between 2 integers inclusive
# sum all the values together and print them out
def print_added_values(val1, val2):
    i = val1
    accum = 0
    while i <= val2:
        print(i)
        accum += i
        i += 1
    print("sum of all values:", accum)


print_added_values(2, 5)

# Output
# 2
# 3
# 4
# 5
# sum of all values: 14
```

An accumulator allows us to store a value over the duration of the while loop

While

Few more examples:

```
# print values between 2 integers inclusive
# sum all the values together and print them out
def print_added_values(val1, val2):
    i = val1
    accum = 0
    while i <= val2:
        print(i)
        accum += i
        i += 1
    print("sum of all values:", accum)


print_added_values(2, 5)

# Output
# 2
# 3
# 4
# 5
# sum of all values: 14
```

An accumulator allows us to store a value over the duration of the while loop

While

Your Turn – Calculate the answer

```
x = 6
while x < 9:
    print(x)
    x = x + 2
print(x)
while x > 7:
    print(x)
    x = x - 1
```

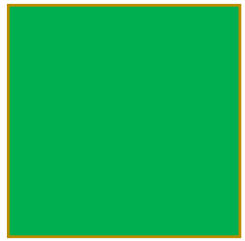
While

Your Turn – Calculate the answer

```
x = 6
while x < 9:
    print(x)
    x = x + 2
print(x)
while x > 7:
    print(x)
    x = x - 1
```

```
6
8
10
10
9
8
```

While

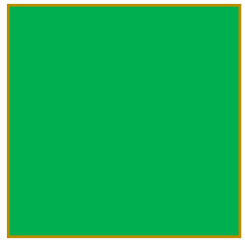


Few more examples:

```
x = 6
while x < 9:
    print(x)
    x = x + 2
print(x)
while x > 7:
    print(x)
    x = x - 1
```

```
6
8
10
10
9
8
```

While



Your Turn – Calculate the answer

```
def puzzle(n):
    v1 = 2
    v2 = 1
    while n > 0:
        if (n % 2 == 1):
            v2 = v1 * v2
        v1 = v1 * v1
        n = n // 2
    return v2


print(puzzle(10))
```

While

Your Turn – Calculate the answer

```
def puzzle(n):
    v1 = 2
    v2 = 1
    print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    while n > 0:
        if (n % 2 == 1):
            v2 = v1 * v2
        v1 = v1 * v1
        n = n // 2
        print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    return v2


print(puzzle(10))
```

While

```
n = 10, v1 = 2, v2 = 1
n = 5, v1 = 4, v2 = 1
n = 2, v1 = 16, v2 = 4
n = 1, v1 = 256, v2 = 4
n = 0, v1 = 65536, v2 = 1024
1024
```

Your Turn – Calculate the answer

```
def puzzle(n):
    v1 = 2
    v2 = 1
    print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    while n > 0:
        if (n % 2 == 1):
            v2 = v1 * v2
        v1 = v1 * v1
        n = n // 2
        print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    return v2


print(puzzle(10))
```

While

```
n = 10, v1 = 2, v2 = 1
n = 5, v1 = 4, v2 = 1
n = 2, v1 = 16, v2 = 4
n = 1, v1 = 256, v2 = 4
n = 0, v1 = 65536, v2 = 1024
1024
```

Your Turn – Calculate the answer

```
def puzzle(n):
    v1 = 2
    v2 = 1
    print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    while n > 0:
        if (n % 2 == 1):
            v2 = v1 * v2
        v1 = v1 * v1
        n = n // 2
        print("n = %d, v1 = %d, v2 = %d" % (n, v1, v2))
    return v2


print(puzzle(10))
```

While Recap

We saw lists are a powerful dynamic format in Python

While Recap

We saw lists are a powerful dynamic format in Python

But they can be slow to populate line by line

While Recap

We saw lists are a powerful dynamic format in Python

But they can be slow to populate line by line

Boolean logic helps us define True or False statements

While Recap

We saw lists are a powerful dynamic format in Python

But they can be slow to populate line by line

Boolean logic helps us define True or False statements

We can use if conditional statements to control code flow

While Recap

We saw lists are a powerful dynamic format in Python

But they can be slow to populate line by line

Boolean logic helps us define True or False statements

We can use if conditional statements to control code flow

Creating a loop in the code can help to reduce workload

While Recap

We saw lists are a powerful dynamic format in Python

But they can be slow to populate line by line

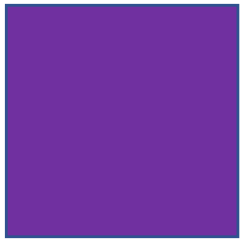
Boolean logic helps us define True or False statements

We can use if conditional statements to control code flow

Creating a loop in the code can help to reduce workload

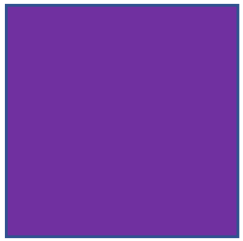
When we have a lot of repeating code

While Recap



We introduced While loops as a mechanism for repeating code

While Recap



We introduced While loops as a mechanism for repeating code

We use While loop when we don't know how long the loops will last

While Recap

We introduced While loops as a mechanism for repeating code

We use While loop when we don't know how long the loops will last

While the condition remains True we continue to execute the statement block of code

While Recap

We introduced While loops as a mechanism for repeating code

We use While loop when we don't know how long the loops will last

While the condition remains True we continue to execute the statement block of code

But… we need some mechanism to make the condition false, otherwise we loop forever…

While Recap

Counters are very important to While loops

While Recap

Counters are very important to While loops

They not only allow us to loop in the While a certain number of times

While Recap

Counters are very important to While loops

They not only allow us to loop in the While a certain number of times

But they also allow us to use list indexing to gather and set information

While Recap

Counters are very important to While loops

They not only allow us to loop in the While a certain number of times

But they also allow us to use list indexing to gather and set information

Incorrect counting can cause the majority of the bugs seen in While loops

While

But what happens when we get caught in an infinite loop

While

But what happens when we get caught in an infinite loop

What can we do to get out of the loop

While

But what happens when we get caught in an infinite loop

What can we do to get out of the loop

And is there anything we can do when the loop completes

While

But what happens when we get caught in an infinite loop

What can we do to get out of the loop

And is there anything we can do when the loop completes

For this we investigate

While

But what happens when we get caught in an infinite loop

What can we do to get out of the loop

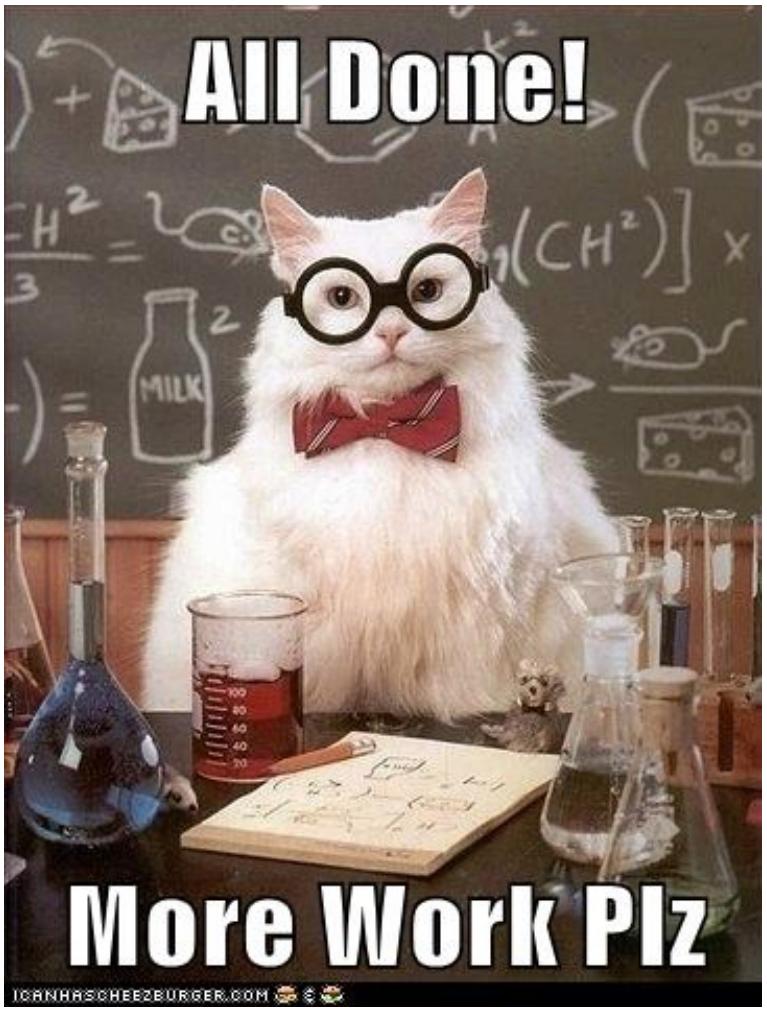
And is there anything we can do when the loop completes

For this we investigate

continue, break and else (yes else – from if/else)

Canvas Student App

Let’s Sign into this lecture now



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

While

We have seen what happens with correct code

```
i = 0
while i < 10:
    print(i)
    i += 1

print("Phew. The While has stopped")

# Output
# 0
# 1
# 2
# 3
# 4
# 5
# 6
# 7
# 8
# 9
# Phew. The While has stopped
```

While

And we have seen what happens when we forget the counter

```
i = 0
while i < 10:
    print(i)
    # i += 1
```

If we forget to increment our count

```
0
0
0
0
0
0
0
0
0
0
0
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While

Let’s look at the counter example a little more

```
# Output
# start time: 1570257161.6979342
# 0
# current time: 1570257161.698022
# 1
# current time: 1570257161.698037
# ...
# 8
# current time: 1570257161.6981199
# 9
# current time: 1570257161.698132
# Phew. The While has stopped
```

While

Let’s look at the counter example a little more

```
# Output
# start time: 1570257161.6979342
# 0
# current time: 1570257161.698022
# 1
# current time: 1570257161.698037
# ...
# 8
# current time: 1570257161.6981199
# 9
# current time: 1570257161.698132
# Phew. The While has stopped
```

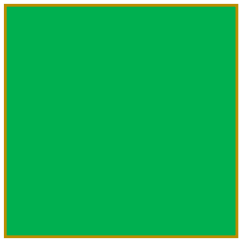
We started @ .6979 and ended @ .6981

While

If we comment out the counter incrementing

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)

print("Phew. The While has stopped")
```

While



And it goes mad….

```
current time: 1570258520.944745
0
current time: 1570258520.944753
0
current time: 1570258520.9447591
0
```

While

And it goes mad….

```
current time: 1570258520.944745
0
current time: 1570258520.944753
0
current time: 1570258520.9447591
0
```

I couldn’t get the start time as my computer becomes unresponsive

Image shows time and zero value for ‘i’

While

So what do we do???

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While

We add a Boolean check

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While

We add a Boolean check

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While

We add a Boolean check

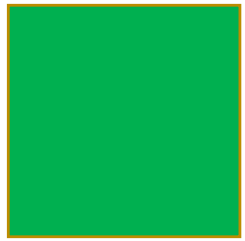
```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While

We add a Boolean check

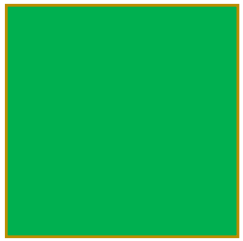
```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While



break

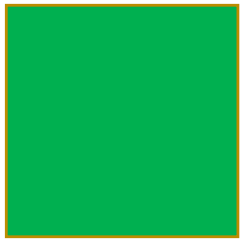
While



break

Is a Python mechanism for stopping loops

While

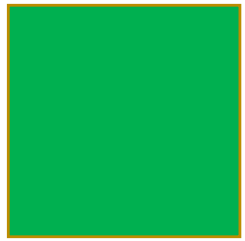


break

Is a Python mechanism for stopping loops

It terminates the execution of the statement block code in the loop

While



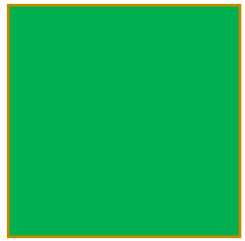
break

Is a Python mechanism for stopping loops

It terminates the execution of the
statement block code in the loop

And Python moves out of the while loop and
to the next location in the code

While



break

Is a Python mechanism for stopping loops

It terminates the execution of the
statement block code in the loop

And Python moves out of the while loop and
to the next location in the code

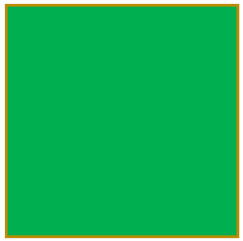
```
print("Phew. The While has stopped")
```

While

We add a Boolean check

```
import time
i = 0
# time in seconds since January 1, 1970 - when time began
# known as "epoch date" - unix start of time
start_time = time.time()

print("start time:", start_time)
while i < 10:
    print(i)
    # i += 1
    # get the current time
    current_time = time.time()
    print("current time:", current_time)
    # if the current time less start time is greater than 3 seconds
    if current_time - start_time > 3:
        print("forced end time:", time.time())
        print("break")
        # break - stops the while loop and continues to next line of code
        break

print("Phew. The While has stopped")
```

While



Let’s look at output

```
# Output
# start time: 1570258881.0076082
# 0
# current time: 1570258881.007679
# 0
# current time: 1570258881.007693
# ...
# 0
# current time: 1570258884.007602
# 0
# current time: 1570258884.007634
# forced end time: 1570258884.007642
# break
# Phew. The While has stopped
```

While

Let’s look at output

```
# Output
# start time: 1570258881.0076082
# 0
# current time: 1570258881.007679
# 0
# current time: 1570258881.007693
# ...
# 0
# current time: 1570258884.007602
# 0
# current time: 1570258884.007634
# forced end time: 1570258884.007642
# break
# Phew. The While has stopped
```

While

Let’s look at output

```
# Output
# start time: 1570258881.0076082
# 0
# current time: 1570258881.007679
# 0
# current time: 1570258881.007693
# ...
# 0
# current time: 1570258884.007602
# 0
# current time: 1570258884.007634
# forced end time: 1570258884.007642
# break
# Phew. The While has stopped
```

While

Let’s look at output

```
# Output
# start time: 1570258881.0076082
# 0
# current time: 1570258881.007679
# 0
# current time: 1570258881.007693
# ...
# 0
# current time: 1570258884.007602
# 0
# current time: 1570258884.007634
# forced end time: 1570258884.007642
# break
# Phew. The While has stopped
```

While

Let’s look at output

```
# Output
# start time: 1570258881.0076082
# 0
# current time: 1570258881.007679
# 0
# current time: 1570258881.007693
# ...
# 0
# current time: 1570258884.007602
# 0
# current time: 1570258884.007634
# forced end time: 1570258884.007642
# break
# Phew. The While has stopped
```

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh