CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

WHILE Loop Examples and Loop Tracing

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

What will we Cover?

- Iteration
- Loop tracing
- Using WHILE loops

CITY
UNIVERSITY OF LONDON
EST 1894

Loop Trace Example

Using a WHILE loop to calculate the **n** factorial

- Factorial of a number **n** is **1*2*…*n**
- At every iteration of the loop multiply the running total by count

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

user input **n** is 3

```
n = 3
total = ?
count = ?
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

Before the loop begins variables are assigned to

```
  n = 3
total = 1
count = 1
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

`1 <= 3` evaluates to `True`

```
n = 3
total = 1
count = 1
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

total = 1*1 = 1

n = 3
total = 1
count = 1

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

`count = 1+1 = 2`

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

```
n = 3
total = 1
count = 2
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

2 <= 3 evaluates to True

n = 3
total = 1
count = 2

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

total = 1*2 = 2

n = 3
total = 2
count = 2

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

count = 2+1 = 3

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

```
n = 3
total = 2
count = 3
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

3 <= 3 evaluates to True

```
n = 3
total = 2
count = 3
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

total = 2*3 = 6

n = 3
total = 6
count = 3

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

count = 3+1 = 4

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

```
n = 3
total = 6
count = 4
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

4 <= 3 evaluates to False

The body of the loop is not executed

```
n = 3
total = 6
count = 4
```

Loop Trace Example

```
n = int(input("enter n: "))
total = 1
count = 1

while count <= n:
    total *= count
    count += 1

print("factorial of", n, "is", total)
```

```
enter n: 3
factorial of 3 is 6
```

```
n = 3
total = 6
count = 4
```

Print the value for the factorial of n

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

Before the loop begins `count` is set to 0

user input **y** is 13

```
y = 13
count = 0
```

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`13 > 1` evaluates to `True`

`y = 13`
`count = 0`

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`y = 13//2 = 6`

`count = 0+1 = 1`

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

```
y = 6
count = 1
```

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`6 > 1` evaluates to `True`

`y = 6`
`count = 1`

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`y = 6//2 = 3`

`count = 1+1 = 2`

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

```
y = 3
count = 2
```

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`3 > 1` evaluates to `True`

`y = 3`
`count = 2`

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

`y = 3//2 = 1`

`count = 2+1 = 3`

The end of loop body is reached, so execution jumps back to the beginning of `while` to test the condition again

`y = 1`
`count = 3`

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 1
3
```

`1 > 1` evaluates to `False`

The body of the loop is not executed

```
y = 1
count = 3
```

Loop Trace Example II

```
count = 0
y = int(input("enter y: "))

while y > 1:
    y //= 2
    count += 1

print(count)
```

```
enter y: 13
3
```

y = 1
count = 3

Try It Yourself

Trace the execution of the following WHILE loop example

```
m = int(input("enter m: "))
n = int(input("enter n: "))
total = 0

while m <= n:
    total += m
    m += 1

print("the total is", total)
```