# While Loops

Making Code Repeat

# Why Do We Need Loops?

Imagine calculating tips for 100 restaurant bills (for simplicity, let's say each bill is $25 with a 20% tip):

```
total = 0
total = total + (25 * 1.20)  # Bill 1
print(f"Total after Bill 1: {total}")

total = total + (25 * 1.20)  # Bill 2
print(f"Total after Bill 2: {total}")

total = total + (25 * 1.20)  # Bill 3
print(f"Total after Bill 3: {total}")

# ... 97 more times?!
```

```
Total after Bill 1: 30.0
Total after Bill 2: 60.0
Total after Bill 3: 90.0
```

Loops let us repeat code efficiently!

# The `while` Loop

Repeats code while a condition is True:

```
count = 1
while count <= 5:
    print(f"Count is {count}")
    count = count + 1

print("Done!")
```

```
Count is 1
Count is 2
Count is 3
Count is 4
Count is 5
Done!
```

Note how in this example:

- The loop runs as long as `count <= 5` evaluates to `True`
- We update `count` inside the loop to eventually stop it

# `while` Loop Structure

```
while condition:
    # Code to repeat
    # Must eventually make condition False!
```

The loop keeps running as long as the condition is `True`

```
temperature = 100
while temperature > 75:
    print(f"Temperature is {temperature}°F - still too hot!")
    temperature = temperature - 5

print(f"Finally cooled down to {temperature}°F")
```

```
Temperature is 100°F - still too hot!
Temperature is 95°F - still too hot!
Temperature is 90°F - still too hot!
Temperature is 85°F - still too hot!
Temperature is 80°F - still too hot!
Finally cooled down to 75°F
```

# `while` Loop vs `if`

Repeats code **while** a condition is True:

```
count = 1
while count <= 5:
    print(f"Count is {count}")
    count = count + 1

print(f"Count is now {count}")
print("Done!")
```

```
Count is 1
Count is 2
Count is 3
Count is 4
Count is 5
Count is now 6
Done!
```

Executes once **if** the condition is True:

```
count = 1
if count <= 5:
    print(f"Count is {count}")
    count = count + 1

print(f"Count is now {count}")
print("Done!")
```

```
Count is 1
Count is now 2
Done!
```

# Counting with `while`

We can use `while` to count up or down:

## 1 to 5 (exit when 6)

```
print("Counting up starting from 1:")
number = 1
while number <= 5:
    print(f"Number is {number}")
    number = number + 1
print(f"Number is now {number}")
```

```
Counting up starting from 1:
Number is 1
Number is 2
Number is 3
Number is 4
Number is 5
Number is now 6
```

## 0 to 4 (exit when 5)

```
print("Counting up starting from 0:")
number = 0
while number < 5:
    print(f"Number is {number}")
    number = number + 1
print(f"Number is now {number}")
```

```
Counting up starting from 0:
Number is 0
Number is 1
Number is 2
Number is 3
Number is 4
Number is now 5
```

## 5 to 1 (exit when 0)

```
print("Counting down from 5:")
countdown = 5
while countdown > 0:
    print(f"{countdown}...")
    countdown = countdown - 1
print("Blast off! 🚀")
```

```
Counting down from 5:
5...
4...
3...
2...
1...
Blast off! 🚀
```

# Summing Bills with `while`

```
total = 0
total = total + (25 * 1.20)  # Bill 1
print(f"Total after Bill 1: {total}")

total = total + (25 * 1.20)  # Bill 2
print(f"Total after Bill 2: {total}")

total = total + (25 * 1.20)  # Bill 3
print(f"Total after Bill 3: {total}")

print("Done!")
```

```
Total after Bill 1: 30.0
Total after Bill 2: 60.0
Total after Bill 3: 90.0
Done!
```

```
bill_count = 1
total = 0
while bill_count <= 3:
    bill = 25 * 1.20  # 20% tip
    total = total + bill
    print(f"Total after Bill {bill_count}: {total:.2f}")
    bill_count = bill_count + 1

print("Done!")
```

```
Bill 1: Total so far = $24.00
Bill 2: Total so far = $48.00
Bill 3: Total so far = $72.00
Bill 4: Total so far = $96.00
Bill 5: Total so far = $120.00
Final total: $120.00
Total after Bill 1: 30.00
Total after Bill 2: 60.00
Total after Bill 3: 90.00
Done!
```

# Accumulating with `while`

Keep a running total:

```
total = 0
count = 1

while count <= 5:
    bill = 20  # Each bill is $20
    tip = bill * 0.20
    total = total + bill + tip
    print(f"Bill {count}: Total so far = ${total:.2f}")
    count = count + 1

print(f"Final total: ${total:.2f}")
```

```
Bill 1: Total so far = $24.00
Bill 2: Total so far = $48.00
Bill 3: Total so far = $72.00
Bill 4: Total so far = $96.00
Bill 5: Total so far = $120.00
Final total: $120.00
```

# Infinite Loops - Be Careful!

If the condition never becomes False:

```
# DON'T RUN THIS!
x = 1
while x > 0:
    print("Help, I'm stuck!")
    # x never changes, so this runs forever!
```

**Always make sure something in your loop will eventually make the condition False!**

# Breaking Out Early with `break`

Sometimes we want to exit a loop before the condition is False:

```
number = 1
while number <= 100:
    if number % 7 == 0:
        print(f"Found a number divisible by 7: {number}")
        break  # Exit the loop immediately
    print(f"Rechecking with {number}...") # This doesn't run after break
    number = number + 1

print("Done searching!")
```

```
Rechecking with 1...
Rechecking with 2...
Rechecking with 3...
Rechecking with 4...
Rechecking with 5...
Rechecking with 6...
Found a number divisible by 7: 7
Done searching!
```

# Input Validation with `while` and `if`/`elif`/`else` and `break`

Sometimes you want a user to give you a specific input, like "yes" or "no".

We can use a `while` loop to keep asking until they meet some criteria, which we will enforce with `if` statements:

```
while True:
    answer = input("Do you like Python? (yes/no): ")
    if answer == "yes":
        print("Great choice!")
        break
    elif answer == "no":
        print("Give it a chance!")
        break
    print("Please type 'yes' or 'no'")
```

This loops forever until we `break` out!

# Input Validation with `while`

We can expand upon this to support other kinds of input validation, like checking if a number is within a range:

```
age = -1  # Start with invalid value

while True:
    age = int(input("Enter your age (0-150): "))
    if age > 0 and age < 150:
        break
    print("That's not a valid age! Try again.")

print(f"Thanks! You are {age} years old.")
```

# Password Example with Attempts

Limit the number of tries:

```
password = "secret123"
attempts = 0
max_attempts = 3

while attempts < max_attempts:
    guess = input("Enter password: ")
    attempts = attempts + 1

    if guess == password:
        print("Access granted!")
        break
    else:
        remaining = max_attempts - attempts
        if remaining > 0:
            print(f"Wrong! {remaining} attempts left.")
        else:
            print("Sorry, no more attempts. Access denied.")
            break
```

# Exercise

bigd103.link/while-loop-calculator