If-Statements and While-Loops

CS for Social Good

Recap of if-else

Last time we went over if-else, the first of several conditional statements that we will cover in this curriculum.

```
x = 8
if x > 5:
    print("x is greater than 5")
else:
    print("x is smaller than 5")
```

Recap of if-else

Last time we went over if-else, the first of several conditional statements that we will cover in this curriculum.

Will this code block
orrectly handle all inputs?
→

```
x = 8
if x > 5:
    print("x is greater than 5")
else:
    print("x is smaller than 5")
```

The elif-statement

Similar to the if-statement, the elif-statement will have a keyword, a condition, a semicolon at the end of the statement followed by an indented code block.

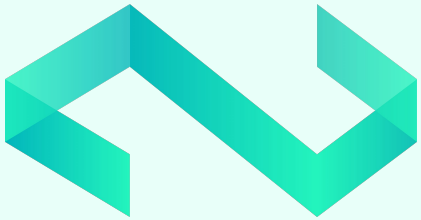
```
x = 5
if x > 5:
    print(“x is greater than 5”)
elif x == 5:
    print(“x is equal to 5”)
else:
    print(“x is smaller than 5”)
```

Structure of the elif-statement

An if/elif/else statement must have exactly one if-statement and an optional else-statement. There is no limit to the number of elif-statements you can put after the initial if-statement.

```
if x == 1:
        print("x is equal to 1")
elif x == 2:
        print("x is equal to 2")
elif x == 3:
        print("x is equal to 3")
elif x == 4:
        print("x is equal to 4")
```

```
if age > 65:
        print("You are a senior citizen")
elif age > 18:
        print("You are an adult")
elif age > 13:
        print("You are an adolescent")
elif age > 1:
        print("You are a child")
else:
        print("You are an infant")
```



Coding Break

While-loops

While-loops are going to be very similar in structure to if-statements, but instead of just executing *if* a statement is true, a while-loop will execute *while* a statement is true.

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :
        print(x)
        x += 1
```

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :          → prints "1"
    print(x)
    x += 1
```

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :          → prints "1"
    print(x)           → prints "2"
    x += 1
```

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :          → prints "1"
    print(x)           → prints "2"
    x += 1             → prints "3"
```

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :          → prints "1"
    print(x)           → prints "2"
    x += 1             → prints "3"
                       → prints "4"
```

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :
    print(x)
    x += 1
```

→ prints “1”
→ prints “2”
→ prints “3”
→ prints “4”
Program is finished

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

What is the value of x when this code block is done?

```
x = 1
while x < 5 :
    print(x)
    x += 1
```

→ prints “1”
→ prints “2”
→ prints “3”
→ prints “4”
Program is finished

The While-loop

Breaking down the code:
If-statements have four main components:

The keyword `**while**`
A **conditional statement** that evaluates to True or False
A colon `**:**` at the end of the line
A **code block** indented one tab further than the if-statement

```
x = 1
while x < 5 :
    print(x)
    x += 1
```

→ prints “1”
→ prints “2”
→ prints “3”
→ prints “4”
Program is finished

Why doesn’t 5 get printed?

Next Time!

For-loops or functions?