If Statements

The *if* statement allows you to write programs that make a choice about which bit of code to run.

Example format:

```
if CONDITION:
    STATEMENTS
```

In this code, if 'CONDITION' evaluates to *True*, the section of code called 'STATEMENTS' will be run.

If 'CONDITION' evaluates to *False*, that section of code will not be run.

If-Else

You can follow an if statement by an else statement, which marks code that will be run if the condition is false:

```
if CONDITION:
    STATEMENTS
else:
    OTHER STATEMENTS
```

In this code, if CONDITION is *True*, STATEMENTS will be run, but if it is *False*, OTHER STATEMENTS will be run instead.

- If and else are both reserved words.

Elif

If you need a choice with three options (e.g. too hot, too cold, and just right), then you nest conditions:

```
if temperature > 60:
    print('too hot')
else:
    if temperature < 40:
        print('too cold')
    else:
        print('just right')
```

Elif is a shorthand for this structure. Wherever you have an *if* immediately followed by an *else*, you can replace the two with an *elif*:

```
if temperature > 60:
    print('too hot')
elif temperature < 40:
    print('too cold')
else:
    print('just right')
```

You can have as many *elif*s as you like, once there is at least one initial *if*.

Syntax Note

To check if two values are the same, you use the '==' operator, as '=' means assignment:

```
if a == b:
    print('a and b have the same value')
```

If Expressions

Similar to If Statements, *If Expressions* are expressions rather than statements – they give back a value rather than doing something.

Format:

```
print('too hot' if t > 60 else 'too cold' if t < 40 else 'just right')
```

- This code is equivalent to the above code in the Elif section.

Handouts & Assignments

- Handout 2: The 'if' Statement
- Assignment 2: The 'if' Statement