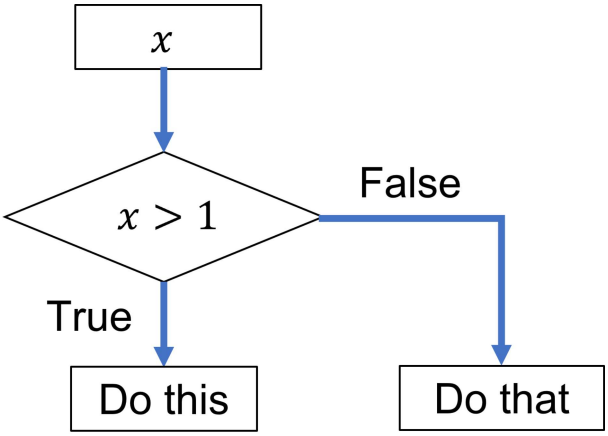
Introduction to Computer Programming

2.1 Control Flow

University of BRISTOL

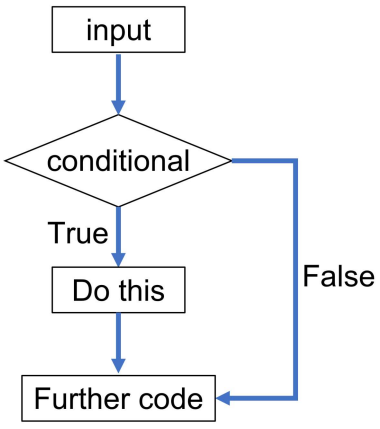
Conditional Statements

- Make decisions within a program and direct the flow.
- Run different blocks of code depending on whether a Boolean expression evaluates to `True` or `False`.
- This decision making is known as **Control Flow**



`if`

Runs a block of code only if a condition is true



In [1]:

```
x = 11

if x > 10:
    print("Do this") # block of code to run only if condition is True

print("Further code")
```

```
Do this
Further code
```

The role of the colon

The colon follows the condition to be evaluated

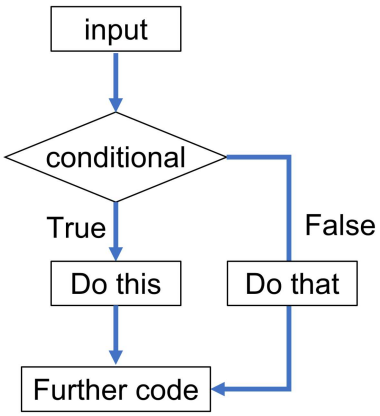
The role of the indent

The indent is used to determine which pieces of code are executed in the case that the condition evaluates to `True` .

The indent can be any number of spaces.
The number of spaces must be the same for all lines in a block of code.
4 spaces is considered best practise.
Many IDEs (e.g. Spyder) automatically indent after you type `if:` .

`if… else`

Runs the indented block of code after `if` if the condition is true.
Otherwise runs the indented block of code after `else`



In [7]:

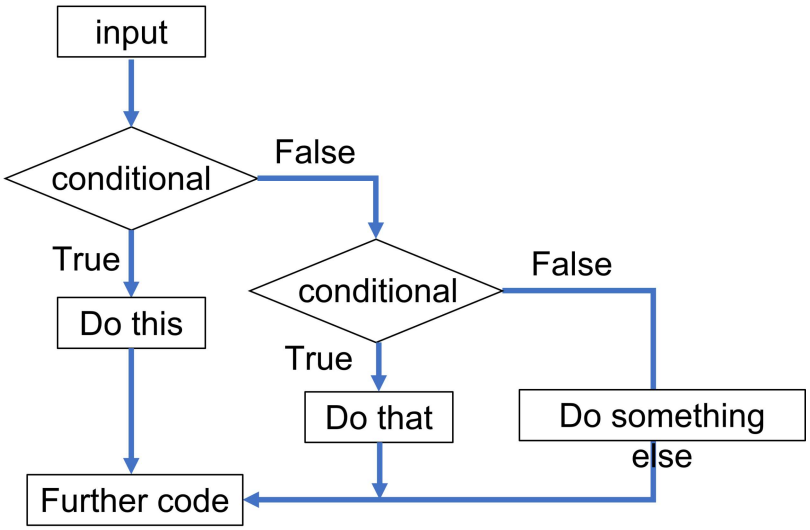
```
x = 5

if x > 10:
    print("Do this") # if condition is True
else:
    print("Do that") # if condition is False

print("Further code")
```

```
Do that
Further code
```

if…elif…(else)

Runs the indented block of code after `if` if the `if` condition is true.
Otherwise runs the indented block of code after `elif` if the `elif` condition is true.
Otherwise runs the indented block of code after `else`

Only one of the three blocks is executed



In [1]:

```
x = 5

if x > 10:
    print("Do this")           # if condition is True
elif x > 5:
    print("Do that")           # if another condition is True
else:
    print("Do something else") # if all preceding conditions are False

print("Further code")
```

```
Do something else
Further code
```

An unlimited numer of `elif` statements can be used after an `if` statement

The `else` statement is optional.

In [12]:

```
x = 5

if x > 10:
    print("x is greater than 10") # if condition is True
elif x > 5:
    print("x is greater than 5") # if another condition is True
elif x > 0:
    print("x is greater than 0") # if another condition is True


print("Further code")
```

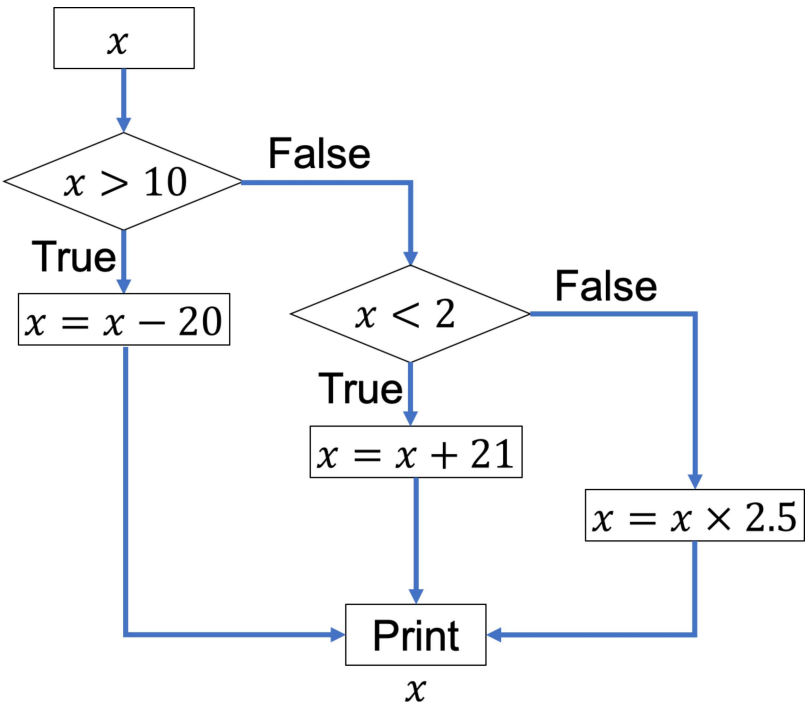
```
x is greater than 0
Further code
```

Summary

- The Python `if` keyword performs a conditional test on an expression for a Boolean (True or False) executes a block of code if the outcome is True.
- Alternatives to an `if` test are provided using `elif` and `else` tests.

In-class Demos

Example 1:
Write a program to modify the initial value of the variable `x` and print the new value, as shown in the flow diagram.



In [20]:

```
# Initial value of x
x = -10.0


# x is greater than 10


# x is less than 2


# x is in range 2 to 10


# Final value of x
print(x)
```

```
Modified x =  -10.0
```

Let's remind ourselves of an example from last week.

> **Example:** Write a program that answers a question based on the current time of day:
> **Is it lunchtime?**
> `True` if between lunch start and end times.
> `False` if not.

In [16]:

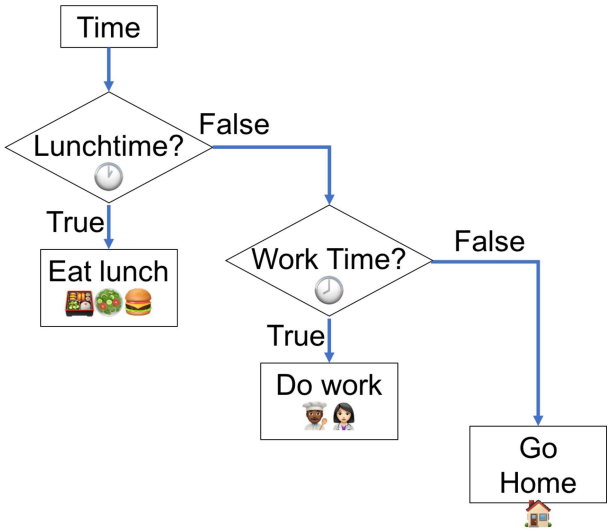
```
time = 9.00          # current time

lunch_starts = 13.00  # time lunch starts
lunch_ends =   14.00  # time lunch ends

lunchtime = time >= lunch_starts and time < lunch_ends

print("Is it lunchtime?")
print(lunchtime)
```

```
Is it lunchtime?
False
```

Let's build on the example from last week by including control statements.

Example 2:
Write a program that tells the user what acitivty to do based on the time of day.

- eat lunch if it is lunchtime
- do work if it is time for work
- go home if it is before or after work



In [7]:

```
# --------- Program from last week ---------

# Variables
t = 9.00            # current time
Ls = 13.00          # lunch starts
Le = 14.00          # lunch ends
Ws = 8.00           # work starts
We = 17.00          # work ends

# lunchtime
lunchtime = Ls <= t < Le

# work_time
work_time = not (   t < Ws      # ... not before work
                 or t > We      # ... or after work
                 or lunchtime) # ... or lunchtime

# ----------------------------------------
```