

# Control statements - intro

Control flow



[https://commons.  
wikimedia.org/wiki/File:  
Modern\\_British\\_LED\\_Traff  
ic\\_Light.jpg](https://commons.wikimedia.org/wiki/File:Modern_British_LED_Traffic_Light.jpg)

# **control statements**

as long as there is a red light, don't drive

If you're late, run

jump up and down ten times

if not enough money in wallet, don't order food

“control flow (or alternatively, flow of control) refers to the specification of the order in which the individual statements, instructions or function calls of an imperative program are executed or evaluated”

[https://en.wikipedia.org/wiki/Control\\_flow](https://en.wikipedia.org/wiki/Control_flow)

# Control statements - if

Control flow



# Assume we're creating a game

In human language:

if health below zero, then the character is dead

In programming:

```
if (health < 0) {  
    die  
}
```

# if

```
if (health < 0) {  
    die  
}
```

**if**

take a step forward

if stone in front of you jump over it

take a step forward

If there actually is a stone there, we will do  
three things (step, jump, step)

.. otherwise we will do two things (step, step)

# if

```
move_forward  
if (stoneAhead) {  
    jump  
}  
move_forward
```

# if - else

If warm put on cap else put on fur hat

# if - else

```
if (health < 0) {  
    die  
} else {  
    scream "ha ha, you aint got me"  
}
```

# **if - else if - if - else**

If warm put on cap, else if hot put on sun hat,  
else put on fur hat

# if - else if - else

```
if (health < 0) {  
    die  
} else if (health < 10) {  
    scream "arrggggg"  
}  
else {  
    scream "ha ha, you aint got me"  
}
```



# if - else if - else

```
if (health < 0) {  
    die  
} else if (health < 10) {  
    scream "arrggggg"  
}  
else {  
    scream "ha ha, you aint got me"  
}
```



# Control statements - while

Control flow

# Go ahead, jump

Jump five times

5 .. 4 .. 3 .. 2 .. 1

# while

Jump five times

```
int jumpsLeft = 5 ;  
while (jumpsLeft>0) {  
    jump;  
    jumpsLeft--;  
}
```

# while

as long as you're not there, keep walking. And  
jump over stones if you find them.

# while

as long as you're not there, take step and if a stone in front of you you should jump over it

# while

```
while (not there) {  
    take a step forward  
    if stone in front of you jump over it  
}
```



# Control statements - for

Control flow

# while - recap

Jump five times

```
int jumpsLeft = 5 ;  
while (jumpsLeft>0) {  
    jump;  
    jumpsLeft--;  
}
```

# while

Very common pattern

```
int jumpsLeft = 5 ;  
while (jumpsLeft>0) {  
    jump;  
    jumpsLeft--;  
}
```

# while → for

Very common pattern

New construct

```
int i = 5;  
while (i>0) {  
    jump;  
    i--;  
}
```

```
for (int i=5;
```

# while → for

Very common pattern

```
int i = 5;  
while (i>0) {  
    jump;  
    i--;  
}
```

New construct

```
for (int i=5; i>0;
```

# while → for

Very common pattern

```
int i = 5;  
while (i>0) {  
    jump;  
    i--;  
}
```

New construct

```
for (int i=5; i>0; i--)
```

# while → for

Very common pattern

```
int i = 5;  
while (i>0) {  
    jump;  
    i--;  
}
```

New construct

```
for (int i=5; i>0; i--) {  
    jump  
}
```

# while → for

Very common pattern

```
int i = 5;
while (i>0) {
    jump;
    i--;
}
```

New construct

```
for (int i=5; i>0; i--) {
    jump
}
```



# for

```
for (int i=5; i>0; i--) {  
    jump  
}
```

Mostly this way

```
for (int i=0; i<5; i++) {  
    jump  
}
```

# Control statements - misc

Control flow

**that's all?**

No, but for now it is.

There's also

do/while, switch/case, jump, goto, .....

.... some other day!!