

**Local variables,
conditionals.**

Variables and Scope

Every variable has a fixed lifetime, called its **scope**.

A field variable exists for the **lifetime of its object**.

A parameter variable only exists **until its method finishes**.

At the end of its scope a variable is **forgotten**.

Local variables

A **local variable** is a variable whose scope is only one method.

Local variables are useful as **temporary space** for performing calculations.

A local variable should always be **initialised**.

Local variables

The syntax for a local variable is:

```
public void method()  
{  
    int localVariable =  
                                initialValue;  
  
    // inside scope  
}  
  
// outside of scope
```

Fields, parameters & local variables

All three kinds of variables:

- store data
- have a type
- have a fixed lifetime

Fields, parameters & local variables

Fields:

- are defined **outside** of methods.
- are initialised in the **constructor**.
- exist until their object is **destroyed**.

Fields, parameters & local variables

Parameters:

- are defined in the **method header**.
- are initialised by the **caller**.
- exist until the method **ends**.

Fields, parameters & local variables

Local variables:

- are defined in the **method body**.
- are initialised **when defined**.
- exist until the **method ends**.

Conditional statements

The syntax of a **conditional statement**:

```
if (condition) {  
    // do this if true  
}  
else {  
    // do this if false  
}  
  
// continue here in either case
```

Conditional statements

Sometimes we need our code to do different things depending on its input.

We use a **conditional statement** (**if statement**) to perform a test and branch based on its results.

Conditional statements

The 'else' part of a conditional is optional:

```
if (condition) {  
    // do this if true  
}  
// continue here in either case
```

Boolean expressions

The condition of an if-statement is a **boolean** value.

A boolean is a **data type** that can have only two values **true** or **false**.

A boolean variable can be declared:

```
boolean waiting = true;
```

Boolean expressions

Example boolean expressions:

```
x == y // equality
```

```
x != y // inequality
```

```
x < y // less than
```

```
x <= y // less or equals
```

Note that we use `==` for **equality**
and `=` for **assignment**.

Printing text

The simplest form of output is **printing** messages to the **console**.

```
System.out.println(  
    "Hello world");
```

We can also print other values:

```
System.out.println(price);
```

Debug printing

One of the most common use of printing is to **debug** your program.

We can print out parts of the program's state and log how they change over time.

Debug printing is useful when your program is long and you don't want to step through the whole thing to find a bug.