

## Loops

While - checks condition, execute code, check again. Once condition fails then exits loop.

Avoid possibility of an 'infinite loop' unless deliberate.  
Beware of loops containing loops - could result in massive cpu/network/file issues.

## For loop

create counting variable, condition, how that changes. 'For' generally neater than 'while'

If you know in advance how many iterations then 'For' is better than 'while'

convention with counting 'i' and if you need a second counter within the loop 'j'

Keeping <sup>'i'</sup> it within the 'For' loop retains control and certainty about its value.

## Arrays

Fundamental collection of variables - a shelf of cups

to declare an array `int [] numberStore`

`int numberStore []`

to instantiate must use 'new'  
index is position with array.

in Java first position is 0

so array size 10 has slots 0-9  
5 has slots 0-4

using indexes (insertion)

```
int [] numStore;
```

declaration

```
numStore = new int [9];
```

instantiation

```
numStore [0] = 77;
```

assignment

see slide!

retrieval

## Array Rules

(references)  
objects or primitives can be held

array is an object, regardless of what it contains

arrays care about type - if you declare int []

then only integers can be contained.

## Iterating over an array

ensure index value is always within known bounds  
- not calling the tenth value in a 9 value array

To avoid this use 'For ~~Each~~ <sup>Each</sup>' loop

But can only be used to 'access' as it is a copy.