Loops

- **if**, **elif**, **else** and **caseof** control in a branching “fashion”
- Also need a way to control how we execute chunks of code multiple times
- Look at two types of loops
  - “**for** loop” - run code a fixed number of times
  - “**while** loop” - run code while / until a condition is met

For loop

- Basic Syntax: **for <variableName> in <iterable type>**
- The variable name is defined and exists in the loop scope (the following indented code)
- The iterable type is anything that we can iterate (“count”) through
  - countup(lo,hi,step) is a procedure that returns such an iterable that goes from lo..hi..step
  - countdown(hi,lo,step) counts down…
  - lo..hi (e.g. 0..10) is shorthand for countup!
  - Other built in and user defined types can be iterable when it makes sense, e.g. strings - the iterator steps through each character
- The block will be repeated with variableName changing according to the iterable each time!

```
# example 1 - numerical range
# note don't have to declare n - its inferred for us
# thing to the right of in denotes a numerical range
for num in 0..10:
  echo num, "(",typeof(num),")"

# we also can iterate to one less than the end - this will more useful than you may expect
for num in 0..<10:
  echo num

# example 2 - range of numbers with a different stride or step-size - can use a procedure countup
for num in countup(0,10,2):
  echo num

# can iterate "backwards" - in other words counting down. Guess what procedure may be called? countdown
# nb step size is still considered "positive"
for num in countdown(10,0,1):
  echo num

# times-table - example of using identifiers from outer scope in the loop
let a = 7
for b in 1..10:
  echo a, "x", b, " = ", a*b
```

While loop

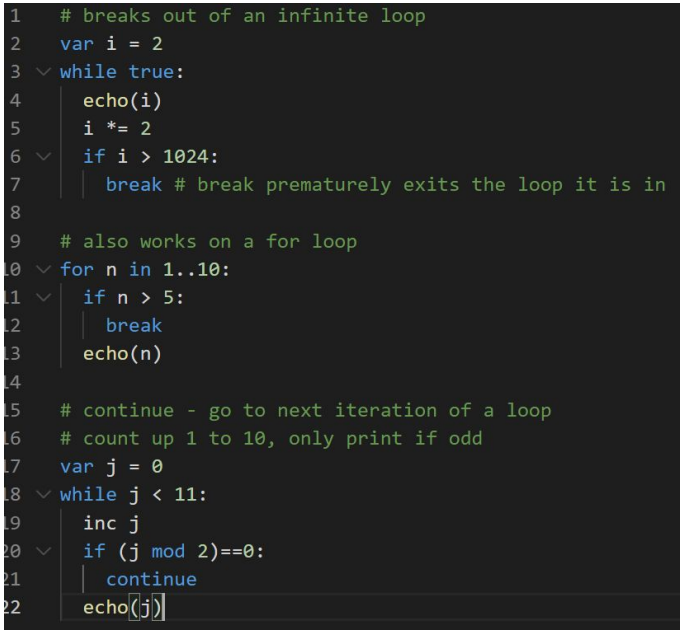
- Similar to for but keep looping while a condition is true
- Syntax is simple

```
while <conditional>:
    do stuff
```

- Conditional can be any expression or variable that evaluates to a bool (true or false)
- Infinite loops possible (by mistake or on purpose)

break and continue

- Fine tune your loops!!!
- break: exits the loop immediately
- continue: immediately go to next iteration of the loop, ignoring any following lines for this step



```
# breaks out of an infinite loop
var i = 2
while true:
  echo(i)
  i *= 2
  if i > 1024:
    break # break prematurely exits the loop it is in

# also works on a for loop
for n in 1..10:
  if n > 5:
    break
  echo(n)

# continue - go to next iteration of a loop
# count up 1 to 10, only print if odd
var j = 0
while j < 11:
  inc j
  if (j mod 2)==0:
    continue
  echo(j)
```