UAX
UNIVERSIDAD ALFONSO X EL SABIO

Condicionales y Bucles

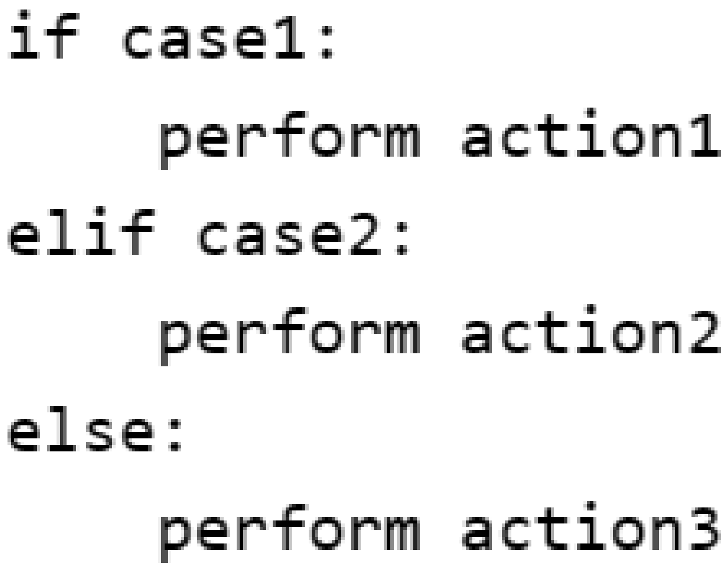
GRADO EN BIOMEDICINA

Bioinformática

idelhgar@uax.es

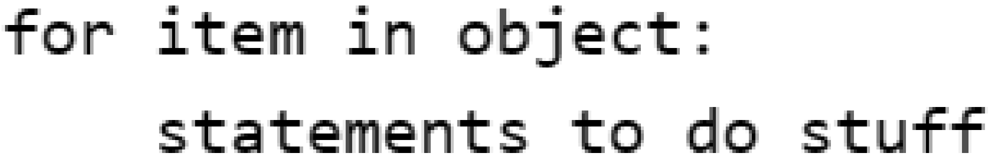
2023

If, elif, else



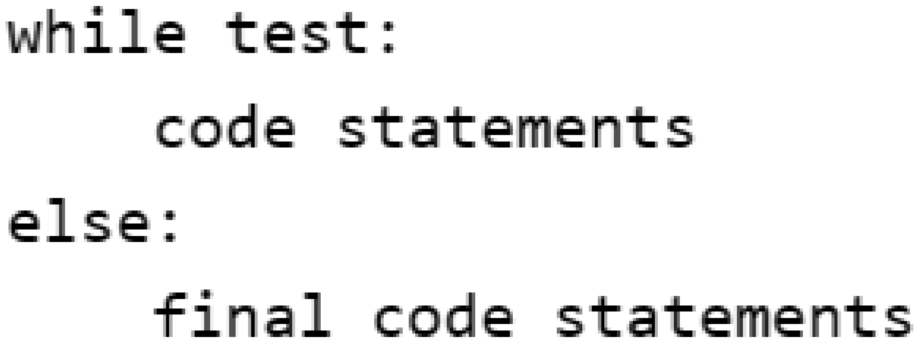
```
if case1:
    perform action1
elif case2:
    perform action2
else:
    perform action3
```

For loop



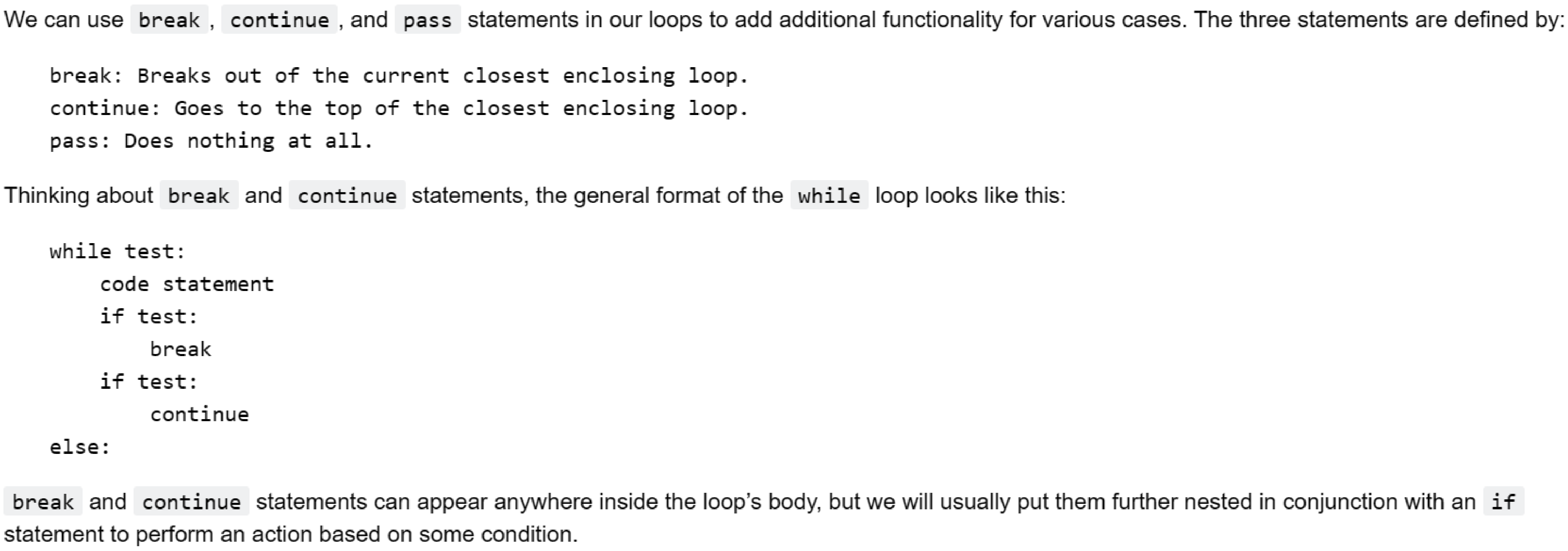
```
for item in object:
    statements to do stuff
```

While loop



```
while test:
    code statements
else:
    final code statements
```

Break, continue, pass



We can use `break`, `continue`, and `pass` statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

Thinking about `break` and `continue` statements, the general format of the `while` loop looks like this:

```
while test:
    code statement
    if test:
        break
    if test:
        continue
else:
```

`break` and `continue` statements can appear anywhere inside the loop's body, but we will usually put them further nested in conjunction with an `if` statement to perform an action based on some condition.

Algunos operadores útiles

- **Range:** genera una lista muy rápida de números enteros.
- **Ennumerate:** se usa sobre todo en los bucles for.
- **In:** se usa durante el bucle for, pero se puede emplear para chequear si un objeto está en una lista.
- **Not in:** para chequear que un elemento NO está en una lista
- **Min y max:** mínimo y máximo de una lista
- **Random:** es una librería de Python que viene por defecto. Genera datos aleatorios.
- **Input:** permite que el usuario introduzca algo