

Loops

```
for i in "list":  
    ≡
```

```
while condition:  
    ≡
```

```
for i in l:  
    l = l + [i]  
    print(i)
```

break

continue

```

def is_prime(n):
    not_prime = False
    if n == 2:
        return True -----> Exit the function
    else: -----> Unnecessary
        for i in range(2, n):
            if n % i == 0:
                break break ← not_prime = True
    * 1. Break
      2. Range exhausted - if not_prime:

```

if _:
return
only if "if" is false

Python solution: for has an "else" !

```
def is_prime(n):
```

```
    if n==2:
```

```
        return True
```

```
    else:
```

```
        for i in range(2,n):
```

```
            if n%i == 0:
```

```
                break
```

```
        else:
```


```
            return True
```

```
    return False
```

executes if for
ends "normally".

reached only if else is not executed

```
def findposition(l, x): # first position of x in l
                       # -1 if x not in l
    for i in range(0, len(l)):
        if l[i] == x:
            return(i)
    else:
        return(-1)
```



Generate all primes upto n

```
def primes(n):
```

```
    primelist = []
```

```
    for i in range(2, n+1):
```

```
        if is_prime(i):
```

```
            primelist = primelist + [i]
```



```
    return(primelist)
```

← make this better

check
factors

only in
primelist

```
def primes(n):  
    primelist = []  
    for i in range(2, n+1):  
        for p in primelist:  
            if i % p == 0:  
                break  
        else:  
            primelist = primelist + [i]  
    return(primelist)
```



Generate first n primes

```
def nprimes(n):  
    i = 2  
    primelist = []  
    while len(primelist) < n:  
        for p in primelist:  
            if i % p == 0:  
                break  
        else:  
            primelist = primelist + [i]  
        i = i + 1
```

Boolean Conditions

if --- : Evaluate to
True or False

Unnecessary to check $x == \text{True}$

if $x == \text{True}$: Same as if x :

Shortcut

Other values are also interpreted as
True, False inside conditions

0 is False

" is False

[] is False

Everything else is True

if primelist: ↔ if primelist != []:

≡

Sequences

 $x[i]$ $x[i:j]$

Strings

Immutable, made up of characters

Lists

• Mutable, arbitrary values

Tuple

Immutable (fixed length) list

 $y = (7, \text{"hello"})$ $y[0] \dashrightarrow 7$ $y[1] \dashrightarrow \text{"hello"}$ $y[1][2:4] \dashrightarrow \text{"ll"}$

$(x, y, z) = (7, \text{"hello"}, \text{True})$

So what?

$(x, y) = (y, x)$ Exchange x & y

$\text{temp} = x$
 $x = y$
 $y = \text{temp}$

List functions

x in l True if value x appears in l

$l.index(x)$ First position in l where x appears
Error if x not in l

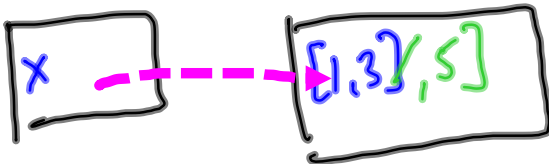
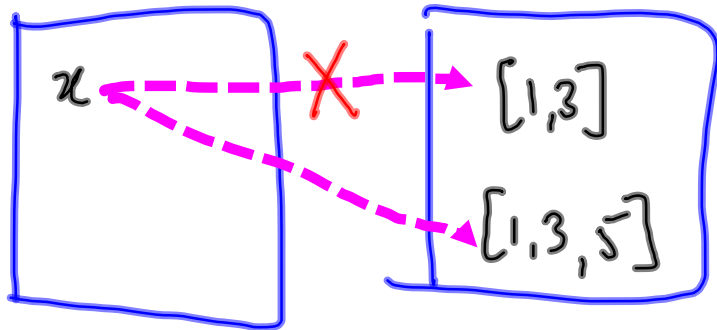
Why $len(l)$, not $l.len()$?

$l.sort()$ Sort l in place

.

$x = [1, 3]$ $x = x + [5]$ $x.append(5)$

append in place

 $x.extend([7, 9])$

$$x = [[1,3], [4,5], [6,7]]$$
$$y = x[1]$$
$$z = x$$
$$x[1] = [8,9]$$
