Mutable vs Immutable

l = [2,3,4]

l[1] = 6 $\dashrightarrow$ [2,6,4]

l[1:3] = [8,9] $\dashleftarrow$ [2,8,9]

l[0:2] = [11,13,15] $\dashrightarrow$ [11,13,15] ✗

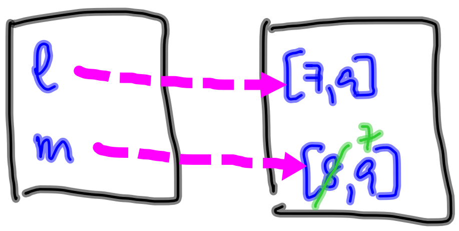
$\dashrightarrow$ [11,13,15,9] ✓

l[1:3] = [6] $\dashrightarrow$ [11,6,9]

l[3:2] $\dashrightarrow$ []

Equality

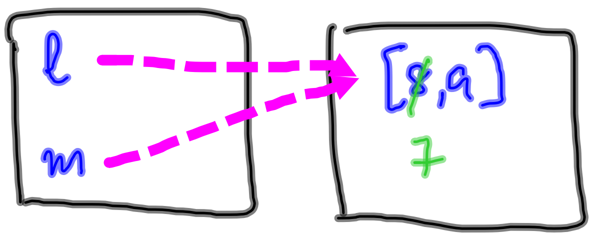
```
l = [7,9]
m = [8,9]
m[0] = 7
```
$\dashrightarrow$ [7,9]



```
if l == m:
    ≡
```

```
l = [8,9]
m = l

m[0] = 7
```



```
if l == m:     True
```

== cannot distinguish "same contents"
from "same object"

Check for "same object"

```
if l is m:
```

True if l,m refer to same objects

Programming language

Control flow

Data manipulation

Basic stmt : Assignment

name = expression

Simplest program

Sequence of assignment statements

Input & output statements

name = input() reads a string from keyboard

print(arg1, arg2, .., argk)

↑ strings non strings converted implicitly using str( )

Conditional execution

```
if conditional-expression :
    stmt1
    stmt2
      ⋮

else:
    —
    —
    —
```

Multiway branch

```
if x<0  do something
else
    if x==0  do something

    else   x>0
             do something
```

Multiway if

```
if x<0:
    ≡
elif x>1:
    ≡
elif x==0:
    ≡
else:        x ∈ (0,1]
    ≡
```

Loop

```
for name in "list" :

for i in range(0,limit):
    if n%i==0 and m%i==0:
        factors = factors + [i]
```

range(i,j) $\dashrightarrow$ [i, i+1, .., j-1]

range(i,j,s) $\dashrightarrow$ [i, i+s, i+2s, .. , i+ks]

s.t $i+(k+1)s \geq j$

$i+ks < j$

Use $i > j$, $s < 0$ to count down

range(8,3,-2) $\dashrightarrow$ [8,6,4]

Typical use of range

l is a list

for i in range(0,len(l))

More loops

```
def gcd(m,n):
    if m%n == 0:
        return(n)
    else
        return(gcd(n, m%n))
```

As a loop

```
m,n
 r = m%n
 if r==0
    return (n)
 else
    m = n
    n = r
```

While loop

```
def gcd(m,n):
   r = m%n
   while r != 0:
      m = n
      n = r
      r = m%n
   return(n)
```

"Efficient" version of inefficient gcd - process factors in reverse order

```
def gcd(m,n):
    limit = min(m,n)
    for i in range(limit,0,-1):
        if m%i==0 and n%i==0:
            largest_factor = i
            break
```

← quit the for loop

~ exit innermost enclosing loop.