

Foundations of Programming

The For, Lists and Tuples

Parameters

Review on parameters

```
def silly(x, y):  
    """ returns x-y """  
    return x-y  
  
x = 21  
y = 2  
z = silly( y, x )  
print(z)
```

What gets displayed when running the program?

- A. 19
- B. -19
- C. 2
- D. 21
- E. Nothing

Review on parameters

```
def silly(x, y):  
    """ returns x-y """  
    return x-y
```

```
x = 21  
y = 2  
z = silly( y, x )  
print(z)
```

Stack frame

y

x

z

y

x

Review on parameters

```
def silly(y):  
    x = y-4  
    return x-y  
  
x = 21  
y = 2  
z = silly( x )  
print(x, ", ", z)
```

What gets displayed when running the program?

- A. 17 , 19
- B. 17 , -4
- C. 21 , 15
- D. 21 , -4
- E. Something else

Review on parameters

```
def silly(y):  
    global x  
    x = y-4  
    return x-y  
  
x = 21  
y = 2  
z = silly( x )  
print(x, ", ", z)
```

What gets displayed when running the program?

- A. 17 , 19
- B. 17 , 15
- C. 21 , 15
- D. 21 , -4
- E. Something else

Loops

Loops

```
for x in [8, 1, 6, 3, 4, 2, 6]:  
    print(x)
```

The loop body will execute one time for each element in the list. Each time through the loop, the loop control variable will take the value of the next element in the list.


```
a = [1,2,1]
for x in a:
    if x > 1:
        print(a)
```

What gets printed by this code?

- A. 2
- B. 1 2
- C. 1 2 1
- D. 2 2 2
- E. Something else

Loops

```
for x in [0, 1, 2, 3, 4, 5, 6]:  
    print(x)
```

```
for x in range(7):  
    print(x)
```

Loops

```
for x in [0, 1, 2, 3, 4, 5, 6]:  
    print(x)
```

```
for x in range(7):  
    print(x)
```

```
list(range(7))          # [0, 1, 2, 3, 4, 5, 6]
```

```
list(range(1, 7))      # [1, 2, 3, 4, 5, 6]
```

```
list(range(0, 7, 2))   # [0, 2, 4, 6]
```

```
for x in range(3,9,3):  
    if x > 3:  
        print(x)
```

What gets printed by this code?

- A. 9
- B. 6 9
- C. 3 6 9
- D. 3 9
- E. 6

```
a = [1,2]
b = [a,a]
for c in b:
    print(c)
```

What gets printed by this code?

- A. c
- B. a a
- C. 1 2 1 2
- D. [1,2] [1,2]
- E. Something else

Lists and Tuples

List

```
a = [1, 3, 5, 9]      # Creates a list 'a'
print(a)             # Prints [1,3,5,9]

b = a[1]             # Select the 1st element from a
print(b)             # Prints 3 (indexing starts at 0!)

b = a[1:3]           # Select from (start) to (end-1)
print(b)             # Prints [3, 5]

a[0] = 7             # Assign to the 0th element of a
a[2:4] = [8,0]       # Assign from (start) to (end-1)
print(a)             # Prints [7,3,8,0]
```

Tuple

```
a = (1, 3, 5, 9)      # Creates a tuple 'a'
print(a)             # Prints (1,3,5,9)

b = a[1]             # Select the 1st element from a
print(b)             # Prints 3 (indexing starts at 0!)

b = a[1:3]           # Select from (start) to (end-1)
print(b)             # Prints (3, 5)

a[0] = 7             # ERROR - Not allowed
a[2:4] = (8,0)       # ERROR - Not allowed
```



```
a = (4,5)
b = [ (1,2) , (4,3) ]
print(b[1])
```

What gets printed by this code?

- A. 1
- B. 1 2
- C. (1,2)
- D. (4,3)
- E. Something else

```
a = (4,5)
b = [ (1,2) , (4,3) ]
print(b[0][1])
```

What gets printed by this code?

- A. This will result in an error
- B. 2
- C. 4
- D. (1,2) (4,3)
- E. Something else

```
sides = [ (0,4), (7,6), (8,3) ]  
for x in sides:  
    if (x[0]-x[1] < 0):  
        print(x)
```

What gets printed by this code?

- A. This will result in an error
- B. -4
- C. 0 7 8
- D. (0,4)
- E. Something else