Fruitful Functions

Chapter 6

Example

```
def square(x):
    print(x*x)

>>> square(4)
16
>>> s = square(4)
16
>>> print(s)
?
```

Example (cont.)

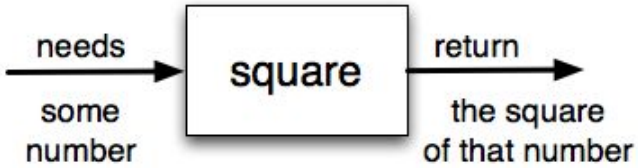
```
>>> print(s)
None

>>> s + 1
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    s + 1
TypeError: unsupported operand type(s) for +: 'NoneType' and
'int'
```

Example (cont.2)

```
def square(x):
    return (x*x)

>>> square(4)
16
>>> s = square(4)
>>> print(s)
16
>>> s + 1
17
```



Return Values

```
def fun(x):
    return x
    print('The end')


def absolute_value(x):
    if x < 0:
        return -x
    if x > 0:
        return x
```

Boolean Functions

```
def is_divisible(x, y):     # name sounds like yes/no
    if x % y == 0:
        return True
    else:
        return False

>>> is_divisible(6, 4)
False
>>> is_divisible(6, 3)
True
```

Boolean Functions (cont.)

```
def is_divisible(x, y):
    return x % y == 0    # more concise solution


if is_divisible(x, y) == True:
    print(' x is divisible by y' )
```

Boolean Functions (cont.2)

```
def is_divisible(x, y):
    return x % y == 0

if is_divisible(x, y) == True:
    print(' x is divisible by y' )

if is_divisible(x, y):
    print(' x is divisible by y' )
```

More Recursion

```
def factorial(n):
    if n == 0:
        return 1
    else:
        recurse = factorial(n-1)
        result = n * recurse
        return result
```

$$0! = 1$$
$$n! = n(n-1)!$$

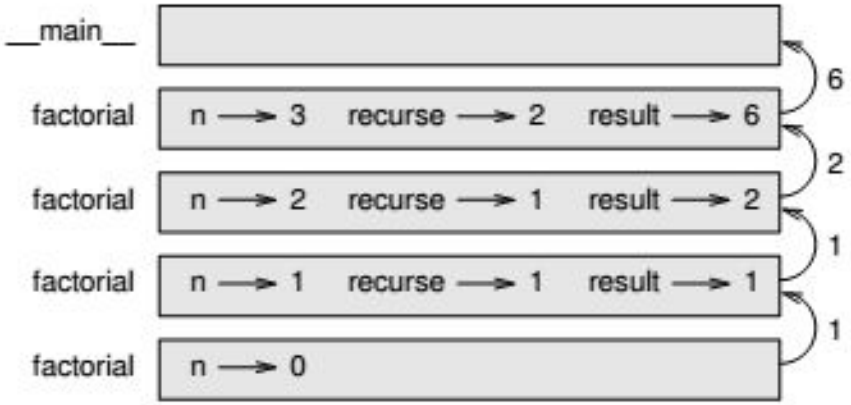
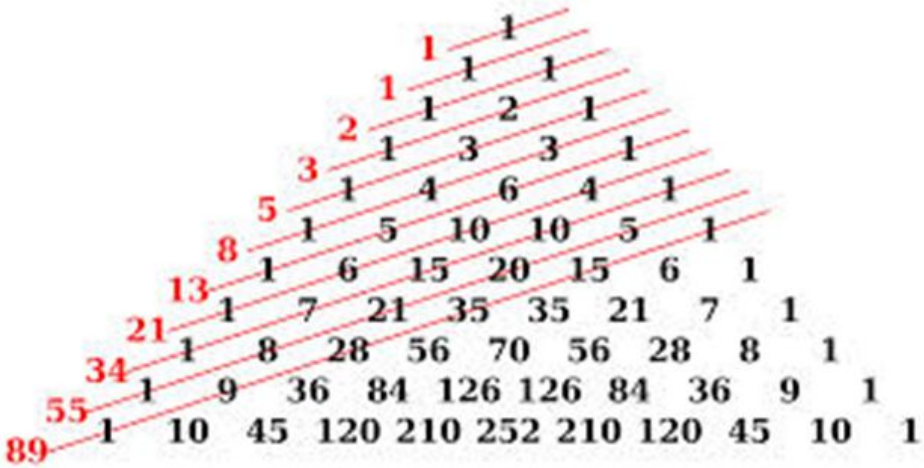
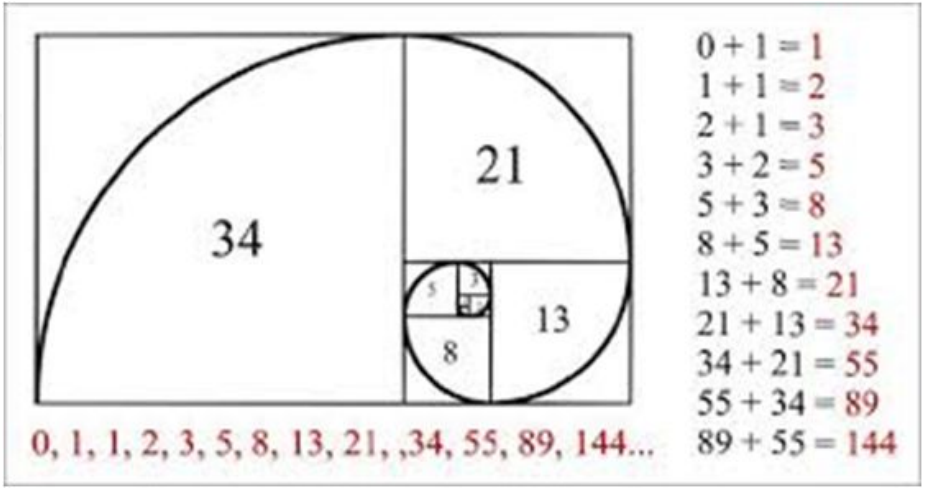


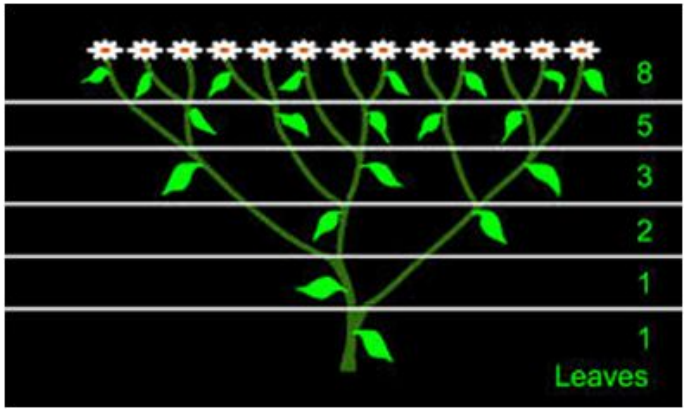
Figure 6.1: Stack diagram.

One More Example





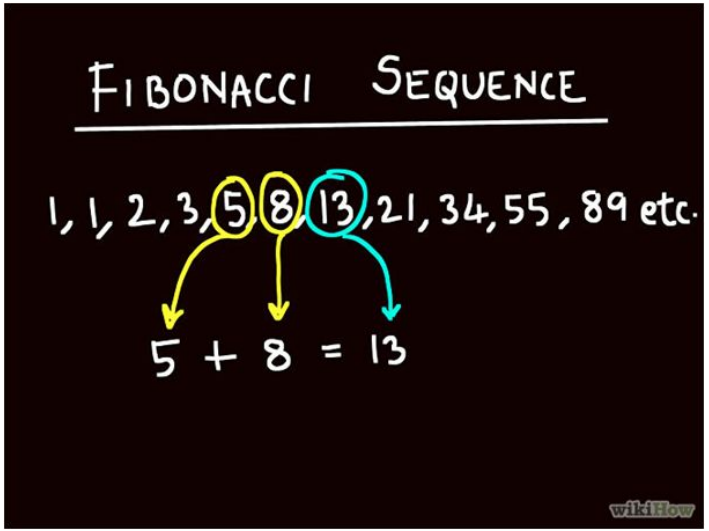
Fibonacci Number



Fibonacci sequence in our hand allows for it to form a perfect curl when we clench our fist.

Fibonacci Number (cont.)

fibonacci (0) = 0
fibonacci(1) = 1
fibonacci(n) = fibonacci(n − 1) + fibonacci(n − 2)




```
def fibonacci (n) :
  if n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    return fibonacci(n-1)+fibonacci(n-2)
```

Checking Types

```
>>> factorial(1.5)
RuntimeError: Maximum recursion depth exceeded

def factorial (n) :
  if not isinstance (n, int):
    print ('Factorial is only defined for integers.')
    return None
...
```