



# Recursion and Global Variables

## Chapter 6

# Recursion

- Self referential
- Who is a “natural born citizen” of the United States?
  1. Any child born inside the United States,
  2. Any child born in wedlock outside the United States both of whose **parents are citizens** of the U.S., as long as one parent has lived in the U.S. prior to the birth of the child, and
  3. Any child born in wedlock outside the United States one of **whose parents is a U.S. citizen** who has lived at least five years in the U.S. prior to the birth of the child, provided that at least two of those years were after the citizen’s fourteenth birthday.

# Re-Recursion

- Base case
  - “A child is born in the United States”
- Inductive case (at least one)
  - Any child born in wedlock outside the United States both of whose parents are citizens of the U.S (*along with some residency requirements*)

# Is a child a US 'birthright' Citizen?

```
def is_usa_citizen(person):  
    if person.birth_country == 'USA':  
        return True  
    elif if (is_usa_citizen(person.parent1) && lived_in_us(person.parent1))  
        || is_usa_citizen(person.parent2) && lived_in_us(person.parent2)):  
        return True  
    else:  
        return (is_usa_citizen(person.parent1) \  
                && weird_residency_test(person.parent1)) || \  
                (is_usa_citizen(person.parent2) \  
                && weird_residency_test(person.parent2))
```

# Factorial

- 10 Factorial or 10!
  - $1*2*3*4*5*6*7*8*9*10$
- Recursive definition
  - $1! = 1$                       base case
  - $n! = n * (n-1!)$       Inductive case

# Factorial Example

## Looping Example

```
def factI(n):  
    """Assumes that n is an int  
> 0  
    Returns n!"""  
    result = 1  
    while n > 1:  
        result = result * n  
        n -= 1  
    return result
```

## Recursion Example

```
def factR(n):  
    """Assumes that n is an int  
> 0  
    Returns n!"""  
    if n == 1:  
        return n  
    else:  
        return n*factR(n - 1)
```

# Fibonacci Numbers

- Breeding like rabbits!
  - This was actually Fibonacci's inspiration
- $f(0) = 1$
- $f(1) = 1$
- $f(n) = f(n-1) + f(n-2)$  – when  $(n > 1)$
- 2 base cases!

# Fibonacci Implementation

```
def fib(n):  
    """Assumes n an int >= 0  
       Returns Fibonacci of n"""  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)
```

- In a turn of events an iterative implementation would be faster
  - More on that when we talk about complexity



# Palindromes

- A word or phrase that reads the same forward or backward
- Examples
  - Racecar
  - Taco cat
  - Amore' Roma
  - Never odd or even
  - Bob
  - A

# Palindromes

- What is the **base case**?
  - String is empty or one character
- What is the **inductive definition**?
  - First and last character are equal
  - The interior string is palindromic

# Global Variables

- Global variables are visible throughout the code
  - Always on the stack
- Should be used carefully and sparingly
- Uses **global** keyword and a variable name
  - Declare at the highest level
  - Use the keyword again to reference the global
- Most commonly used to create a global constant

# Global Variable Example

```
def fib(x):  
    """Assumes x an int >= 0  
        Returns Fibonacci of x"""  
    global numFibCalls  
    numFibCalls += 1  
    if x == 0 or x == 1:  
        return 1  
    else:  
        return fib(x-1) + fib(x-2)  
  
def testFib(n):  
    for i in range(n+1):  
        global numFibCalls  
        numFibCalls = 0  
        print('fib of', i, '=', fib(i))  
        print('fib called', numFibCalls, 'times.')
```