String en Python (parte 1)

Rodrigo Toro Icarte

Comandos importantes

Definición: Un String es una cadena de caracteres.

```
s = "yo soy tu padre"
```

Dentro de un string pueden existir caracteres especiales que permiten incluir saltos de línea (\n), comillas (\" \') y tabuladores (\t).

```
s = "Luke... \n\t \"yo soy tu padre\""
print(s)
# >>> Luke...
#               "yo soy tu padre"
```

Para pedir un string al usuario se utiliza la función *input()*, y para castearlo se usa *int()*, *float()*, *complex()* y *boolean()*.

```
# pedir string
a = input("Ingrese input")

# castear string
a_int = int(a)
a_float = float(a)
a_complex = complex(a)
a_bool = bool(a)
```

La comparación entre dos strings es caracter a caracter (==, ! =, <, <=, >, >=) según formato **ASCII** (orden *casi* alfabético).

```
"hola" == "hola"      # >>> True
"hola" == "oli"       # >>> False
"hola" != "oli"       # >>> True
"hola" < "oli"        # >>> True
"hola" > "a"          # >>> True
```

Podemos acceder a una caracter específico del string mediante su índice, pero no modificarlo.

y	**o**		**s**	**o**	**y**		**t**	**u**		**p**	**a**	**d**	**r**	**e**
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
s = "yo soy tu padre"
print(s[7])   # >>> t
print(s[-8])  # >>> t
s[7] = "s"    # >>> Error!
```

Función *len()* permite conocer el largo de un string (número de caracteres). Así podemos recorrer un string de largo cualquiera.

```
s = input("Ingrese string: ")
j = 0
while(j < len(s)):
  print(s[j])
  j += 1
```

Para recorrer un string también podemos utilizar el comando **for** (que es una alternativa a usar **whiles**).

```
s = input("Ingrese string: ")
for i in s:
  print(i)
```

Finalmente existen 4 operadores básicos para trabajar con strings.

- $a + b \rightarrow$ Concatena a y b.
- n * $a \rightarrow$ Concatena n veces a.
- a **in** $b \rightarrow$ Es True ssi a es parte de b.
- a **not in** $b \rightarrow$ Es True ssi a no es parte de b.

```
a = "hola"; b = "chao"
print(a+b)              # >>> holachao
print(3*a)              # >>> holaholahola
print("ol" in a)        # >>> True
print("ol" not in b)    # >>> True
```

Ejercicios resueltos

1. Cree una función que reciba un string y retorne el mismo string, pero sin los caracteres pares.

   Solución:

```
def quitar_pares(s):
  ret = ""
  i = 0
  while(i < len(s)):
    if(i % 2 == 1):  # solo agrego posiciones impares
      ret += s[i]
    i += 1
  return ret

# llamamos a la función con un string cualquiera
print(quitar_pares("yo soy tu padre"))
```

2. Cree una función que reciba un string y dos enteros i, j, tal que $i \leq j$; y retorne la sub-parte del string que comienza en i y termina en j-1.

   Solución:

```
def substring(s, i, j):
  ret = ""
  while(i < j):
    ret += s[i]
    i += 1
  return ret

# llamamos a la función con un string cualquiera
print(substring("yo soy tu padre",3,6))
```

3. Cree una función que reciba un string y retorne el string invertido.

   Solución:

```
def invertir(s):
  ret = ""       # String nulo!
  for c in s:
    ret = c + ret
  return ret

# llamamos a la función con un string cualquiera
print(invertir("yo soy tu padre"))
```

4. Cree una función que reciba un string **s** y retorne **True** ssi **s** es un palíndromo (una palabra que se lee igual en ambos sentidos, sin considerar espacios).

   Solución:

```
def invertir(s):        # retorna "s" invertido
  ret = ""
  for c in s: ret = c + ret
  return ret

def quitar_espacios(s): # retorna "s" sin espacios
  ret = ""
  for c in s:
    if(c != " "): ret += c
  return ret

def palindromo(s):
  s = quitar_espacios(s)
  return s == invertir(s)

# llamamos a la función con un string cualquiera
print(palindromo("yo soy tu padre"))
print(palindromo("sometamos o matemos"))
```

Ejercicios propuestos

1. Cree una función que resuelve el Capicúa usando strings.

2. Cree una función que retorne el número de palabras presentes en un string (obs: considere que toda palabra válida está separada por un espacio de la anterior).

3. Cree un programa que pida párrafos al usuario hasta que él ingrese un '-1'. Guarde los párrafos en un string (considerando saltos de línea). Al finalizar el programa, muestre al usuario el texto completo ingresado.