

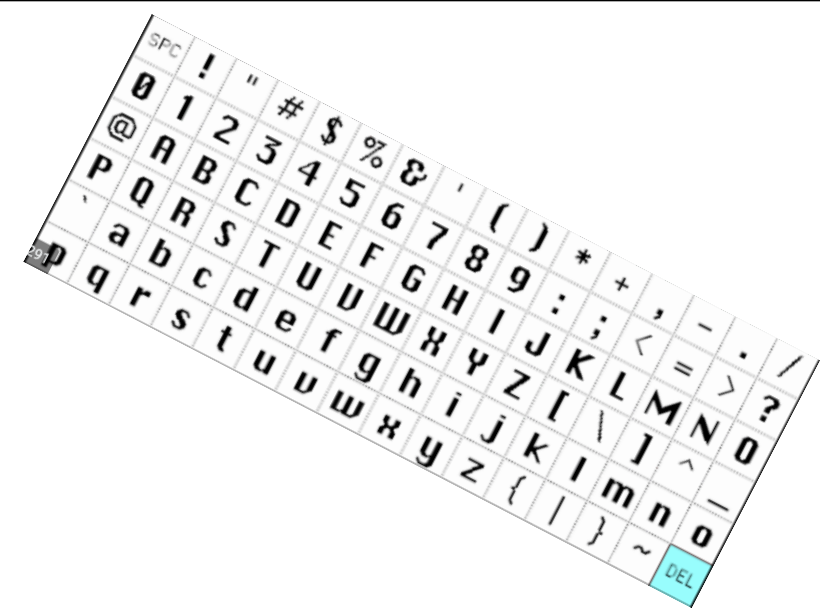
Cadenas

Juan Zamora Osorio

IWI-131 / 2021

Cadenas

- También conocidas como Strings
- Secuencias de caracteres
 - Carácter ~ String de largo 1
- Por ejemplo: "IWI131"
 - Se compone de 6 caracteres: "I", "W", "I", "1", "3", "1"
- Cada carácter es un símbolo proveniente de una tabla
 - ASCII
 - UTF-8



Definiendo cadenas



- Una **palabra** o expresión de **tipo String** (str) se **define encerrando** los caracteres entre comillas (simples o dobles)
- Dentro de cada palabra, sus caracteres tienen ***posiciones relativas*** enumeradas desde 0 hasta la cantidad de caracteres menos 1

Palabra de largo 6

0 1 2 3 4 5
“ I W I 1 3 1 ”

Acceso a caracteres

- Es posible “rescatar” cada caracter de una palabra
 - Usando sus ***posiciones relativas***
 - ...en conjunto con los paréntesis cuadrados []
- Por ejemplo,
 - el **tercer** elemento de “IWI131” tiene la posición relativa **2**
 - Se puede rescatar mediante la expresión “IWI131”[2]
- Al rescatar un(os) carácter(es) es posible
 - Almacenarlos en variables **como** si fuera **palabras nuevas**
 - Operar y crear nuevos Str

La INMUTABILIDAD de los Str

- Inmutable ~ Algo que **no** cambia
- Cada palabra o Str es un objeto que **No** puede ser modificado
 - Reemplazar carácter(es) ✗
 - Eliminar o insertar carácter(es) ✗
- Por ejemplo, intente realizar la siguiente operación
 - "IWI131"[0] = "i"

¿Existe alguna manera de crear/modificar palabras?

¿Existe alguna manera de crear/modificar palabras?

- ¡Varias! ... **No para modificar**, pero si es posible re-utilizar palabras para crear otras nuevas
- Ejecutar y estudiar en detalle el siguiente código

```
>>> p1 = "IWI-131"  
>>> p2 = "iwi" + p1[3] + p1[4] + p1[5] + p1[6]  
>>> p3 = "iwi" + p1[3:7]
```

- El símbolo + es usado para unir o concatenar palabras
- Dentro de [...] pueden indicarse posiciones o rangos de posiciones

¿Existe alguna manera de crear/modificar palabras?

- ¡Varias! ... **No para modificar**, pero si es posible re-utilizar palabras para crear otras nuevas
- Ejecutar y estudiar en detalle el siguiente código

```
>>> p1 = "IWI-131"
>>> p2 = "iwi" + p1[3] + p1[4] + p1[5] + p1[6]
>>> p3 = "iwi" + p1[3:7]
```

Handwritten annotations:

- A bracket above the string "IWI-131" in the first line is labeled with indices 0 1 2 3 4 5 6.
- An arrow points from the bracketed string to the text "Palabra de largo 7".
- An arrow points from the slice notation [3:7] in the third line to the text "Notar que es mayor que la última posición relativa (6)".

- El símbolo + es usado para unir o concatenar palabras
- Dentro de [...] pueden indicarse posiciones o rangos de posiciones

Ejercicios

1. Se define la variable **p1="lhao"**. Use sus caracteres, posiciones relativas y [...] para construir la palabra "hola" .
2. Construya al menos 3 anagramas con los caracteres de la palabra "amor" almacenada en la variable **p**, sus posiciones relativas y los [...] la palabra "amor" almacenada en la variable p
Ejemplo: p[0] + p[3] + p[1] + p[2] permite obtener la palabra "armo"
3. Utilice la función len para obtener la cantidad de caracteres de la palabra "Python es genial!" almacenada en la variable p
4. Utilice de manera combinada la función len junto con los [...] para obtener el último carácter de la palabra "UTFSM"

Iterando sobre Str

- Los caracteres de una palabra pueden ser revisados automáticamente
- Por ejemplo, para crear un programa que cuente la cantidad de vocales en una palabra
- Podemos hacer esto visualmente para un palabra específica
 - Pero, ¿Qué hacemos cuando no conocemos esa palabra?

```
pb = input("Ingrese una palabra:")  
... # resto del código que cuenta vocales  
...
```

Iterando sobre Str

- Usar estructura repetitiva como **patrón de contador**
- Variable itera sobre cada posición relativa

```
pb = input("Ingrese una palabra:")
```

```
i = 0
```

```
largo_palabra = len(pb)
```

```
while i < largo_palabra:
```

```
    ...
```

```
    i = i + 1
```

Operación sobre caracter
en posición i

if pb[i] = "a" or...



Iterando sobre Str

- Usar estructura repetitiva **for**
- Variable de control itera sobre cada posición relativa

```
pb = input("Ingrese una palabra:")  
for x in pb:  
    # valor de x cambia en cada iteración  
    ...
```

Ejercicios

1. Recorra una palabra almacenada en la variable `p`. Cada vez que encuentre una `'r'` muestre un mensaje con la función `print`
2. Haga lo mismo de la pregunta anterior, pero además contabilice la cantidad de `'r's` que aparecen y al finalizar muestre este valor.
3. Utilice el operador `in` para verificar que una palabra está contenida en otra. Por ejemplo, `"ola"` in `"cola"`

- Len
- While
- For
- Upper / lower
- +
- Replace
- in

Ejercicios

- Substrings de la frase: “legando a la sociedad de Valparaíso una Universidad”
- Rutear el siguiente programa:

```
texto = input("Texto: ")
inicio = True
convertido = ""
for c in texto:
    if inicio:
        convertido = convertido + c.upper()
        inicio = False
    elif c == " ":
        inicio = True
    else:
        convertido = convertido + c
print(convertido)
```

Realice el ruteo del siguiente programa e indique qué es lo que imprime. Cada vez que el valor de una variable cambie, escríbalo en una nueva fila de la tabla. Recuerde que si una variable es de tipo STRING, debe colocar su valor entre comillas simples ''. Importante: La tabla tiene suficientes filas.

```
a = "b"
b = 1
s = b + 2
while(b < s and a[-1] != "a"):
    if(b % 2 != 0):
        b += 1
    s += b
    if(s % 4 != 0):
        s -= 1
    else:
        a += "a"
print s
```

[illegible][illegible]

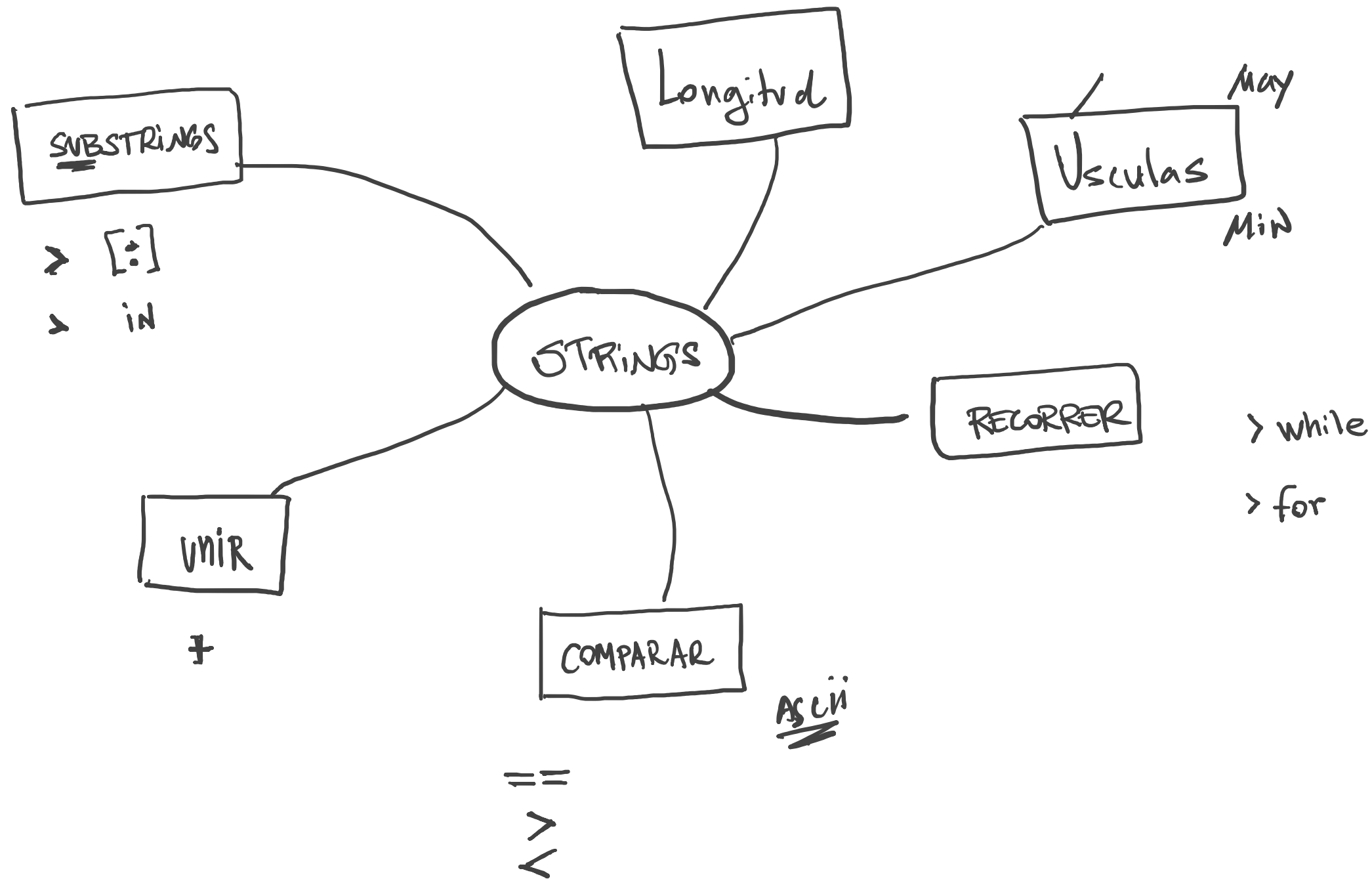
En el básquetbol existen tres diferentes tipos de anotaciones:

- el tiro libre (L), que vale un punto,
- el doble (D), que vale dos puntos, y
- el triple (T), que vale tres puntos.

Un partido de básquetbol está dividido en varios períodos.

Usted debe escribir un programa que reciba como entrada una única línea, que contenga todas las anotaciones realizadas por un equipo de básquetbol durante un partido. Las anotaciones de períodos distintos deben ir separadas por un espacio. Como salida, debe mostrar la cantidad de puntos obtenidos en cada período y los puntos totales, siguiendo el formato del ejemplo.

```
Anotaciones: DDTDLLDD DDLDT TDTLLD DDDDD
15 puntos en el periodo 1
10 puntos en el periodo 2
12 puntos en el periodo 3
10 puntos en el periodo 4
Total: 47 puntos
```

+	0	1	2	3	4	5	6	7	8	9
30				!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			