

STRINGS

## Retake of test

- Format is the same as in the main test
- Best attempt counts
- Possible date and time: Friday, October 18, 14:15-18:00.

# Project

- Register your team
- Submit project description
- Schedule
  - Form a team and select topic: **October 14**
  - Alpha version: **November 11**
  - Beta version: **December 9**

## Test review

Using the functions `add` and `multiply`, rewrite the expression

$$S = a + b * (c + d)$$

in such a way that the result doesn't contain any multiplication signs and any addition signs.

## Test review

The variable `x` has an integer value. Write a piece of Python code that finds out whether `x` is odd or even, and in the latter case, additionally, finds whether `x` is divisible by 4 or not.

```
_____:  
    print("x is odd")  
else:  
    print("x is even, and more precisely:")  
    _____:  
        print("x is even and divisible by 4")  
    _____:  
        print("x even and not divisible by 4")
```

# Test review

- Car worth
- Jubilees

## String data type

```
>>> type("mg5#X4")
```

```
<class 'str'>
```

```
>>> type('mg5#X4')
```

```
<class 'str'>
```

Single or double quotes

## Conversion to numeric type and back

<code>int('12')</code>	<code>-&gt;</code>	<code>12</code>
<code>float('34.5')</code>	<code>-&gt;</code>	<code>34.5</code>
<code>str(12)</code>	<code>-&gt;</code>	<code>'12'</code>
<code>str(34.5)</code>	<code>-&gt;</code>	<code>'34.5'</code>



# Indices

characters	→	C	e	d	a	r
indices	→	0	1	2	3	4

```
word = "Cedar"
```

```
letter = word[1] → e
```

```
print(letter)
```

```
print(word[5]) →
```

```
Traceback (most recent call last):  
print(word[5])  
IndexError: string index out of range
```

# Length

characters → 

C	e	d	a	r
---	---	---	---	---

indices → 

0	1	2	3	4
---	---	---	---	---

```
word = "Cedar"  
length = len(word) → 5  
print(length)
```

# Looping through strings

- Loop over indices

```
word = 'Flick'  
index = 0  
while index < len(word):  
    letter = word[index]  
    print(index, letter)  
    index = index + 1
```

```
0 F  
1 l  
2 i  
3 c  
4 k
```

- Loop over characters

```
word = 'Flick'  
for letter in word:  
    print(letter)
```

```
F  
l  
i  
c  
k
```

# Slicing

characters	→	C	l	e	a	r		c	a	c	h	e
indices	→	0	1	2	3	4	5	6	7	8	9	10

```
>>> text = 'Clear cache'
```

```
>>> text[3:8]
'ar ca'
```

```
>>> text[4:]
'r cache'
```

```
>>> text[:9]
'Clear cac'
```

```
>>> text[:]
'Clear cache'
```

```
>>> text[:-2]
'Clear cac'
```

# Operations with strings

- Concatenation +

```
>>> space = " "  
>>> "Talk" + space + "show"  
'Talk show'
```

- Multiplication \*

```
>>> ("Talk" + space) * 3  
'Talk Talk Talk '
```

# Logical expressions using strings

- Substring

```
>>> "lo" in "Hello"  
True
```

```
>>> "a" in "Hello"  
False
```

- Equality

```
>>> "hello" != "Hello"  
True
```

```
>>> "hello" == "Hello"  
False
```

- Inequality

```
>>> "world" > "Hello"  
True
```

```
>>> "world" < "Hello"  
False
```

# Functions

capitalize	isalnum	join	rsplit
casefold	isalpha	ljust	rstrip
center	isdecimal	lower	split
count	isdigit	lstrip	splitlines
encode	isidentifier	maketrans	startswith
endswith	islower	partition	strip
expandtabs	isnumeric	replace	swapcase
find	isprintable	rfind	title
format	isspace	rindex	translate
format_map	istitle	rjust	upper
index	isupper	rpartition	zfill

<https://docs.python.org/3/library/stdtypes.html#string-methods>