

< Teach
Me
Skills />

Занятие 7.

Исключения. Работа с файлами

1

Исключения

Exceptions

Что это такое?

- еще один тип данных в python
- необходимы для того чтобы сообщать программисту об ошибках
- нужны для того чтобы мы могли некоторые ошибки обработать в коде и при этом наша программа не ломалась

```
try:  
    1 / 0  
except Exception:  
    print("Произошла ошибка!")  
else:  
    print("Выполнюсь только в случае если код в блоке try завершился")  
    print("буз ошибок, return, continue, break")  
finally:  
    print("Выполнюсь в любом случае")
```

Произошла ошибка!

Выполнюсь в любом случае

Как происходит обработка исключений

Нормальное выполнение

```
try:  
    # основные  
    # действия  
  
except exception:  
    # обработка  
    # исключения  
  
finally:  
    # освобождение  
    # ресурсов  
  
    # выполнение  
    # продолжается  
    # здесь
```

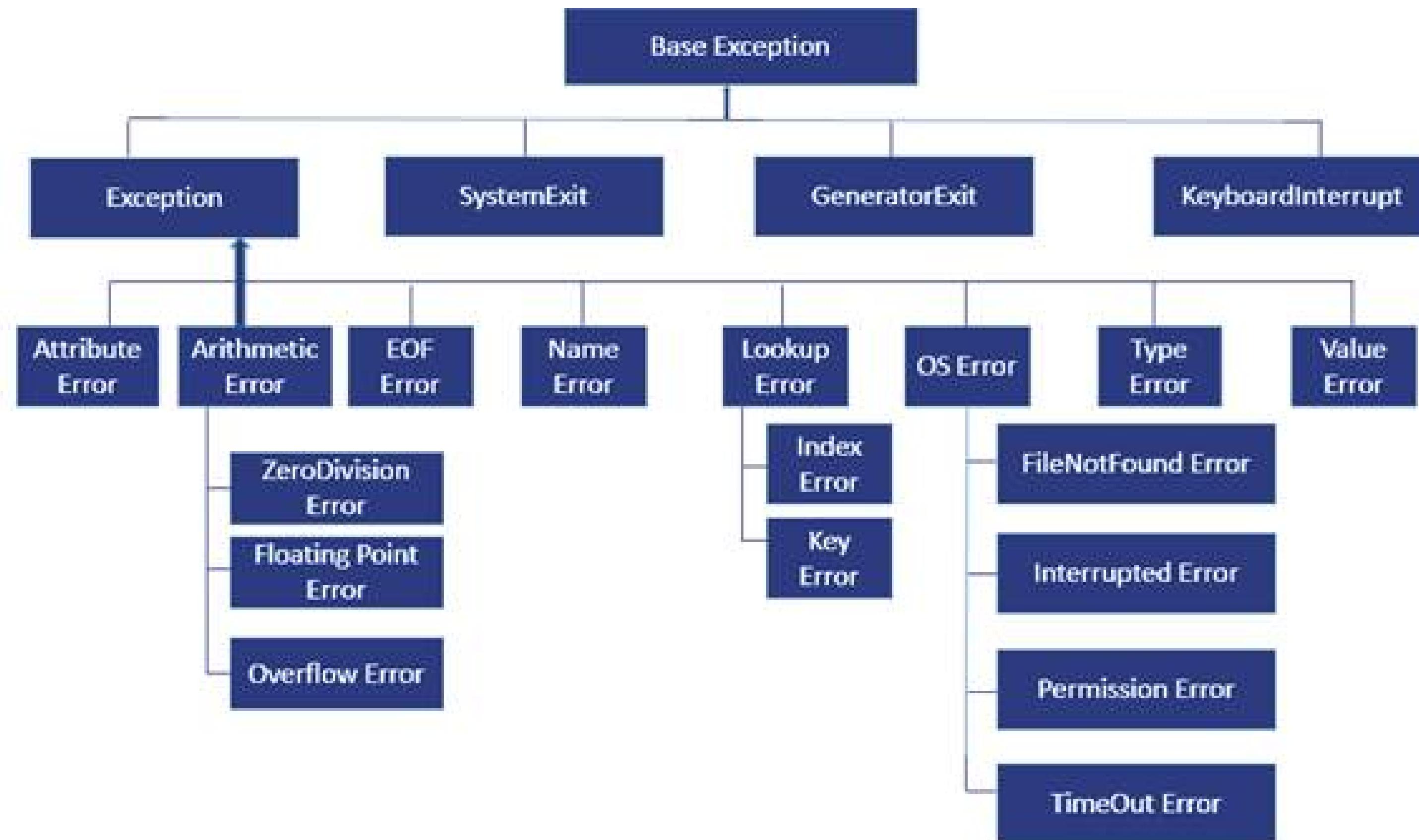
Обработанное исключение

```
try:  
    # основные  
    # действия  
  
except exception:  
    # обработка  
    # исключения  
  
finally:  
    # освобождение  
    # ресурсов  
  
    # выполнение  
    # продолжается  
    # здесь
```

Необработанное исключение

```
try:  
    # основные  
    # действия  
  
except exception:  
    # обработка  
    # исключения  
  
finally:  
    # освобождение  
    # ресурсов  
  
    # исключение  
    # передается вверх  
    # по стеку вызовов
```

Иерархия исключений



Порядок обработки

```
# неправильно
try:
    1 / 0
except Exception:
    print("Произошла ошибка!")
except ZeroDivisionError:
    print("Произошла ошибка деления на 0!")
```

Произошла ошибка!

```
# правильно
try:
    1 / 0
except ZeroDivisionError:
    print("Произошла ошибка деления на 0!")
except Exception:
    print("Произошла ошибка!")
```

Произошла ошибка деления на 0!

Виды ошибок

SystemExit - исключение, порождаемое функцией `sys.exit` при выходе из программы

KeyboardInterrupt - порождается при прерывании программы пользователем (обычно сочетанием клавиш `Ctrl+C`)

GeneratorExit - порождается при вызове метода `close` объекта `generator`.

Exception - а вот тут уже заканчиваются полностью системные исключения (которые лучше не трогать) и начинаются обычновенными которыми можно работать

- **ArithmeticalError** - арифметическая ошибка
 - **ZeroDivisionError** - деление на ноль
- **AttributeError** - объект не имеет данного атрибута (значения или метода)
- **ImportError** - не удалось импортирование модуля или его атрибута
- **LookupError** - некорректный индекс или ключ
 - **IndexError** - индекс не входит в диапазон элементов
 - **KeyError** - несуществующий ключ (в словаре, множестве или другом объекте)
- **NameError** - не найдено переменной с таким именем
- **TypeError** - операция применена к объекту несоответствующего типа
- **ValueError** - функция получает аргумент правильного типа, но некорректного значения

Как создать свое исключение

```
class MyError(Exception):
    def __init__(self, text):
        self.txt = text

a = input("Input positive integer: ")

try:
    a = int(a)
    if a < 0:
        raise MyError("You give negative!")
except ValueError:
    print("Error type of value!")
except MyError as mr:
    print(mr)
else:
    print(a)
```

Input positive integer: -1
You give negative!

Как самостоятельно вызвать исключение

```
: raise ValueError("текст ошибки")
```

Что это такое?

- easier to ask for forgiveness than permission - проще попросить прощение, чем разрешение
- более питонячий вариант
- что является нормальным поведением нашей программы? явное лучше неявного

```
d = {}

try:
    d["key"] # нормальное поведение нашей программы
except KeyError:
    print("нет такого ключа") # исключение (необычное поведение)
```

2

Работа с файлами

Кодировки

Кодировка - соответствие между визуальными символами и числами. Необходимы, так как компьютеры созданы для работы с числами и не понимают текст

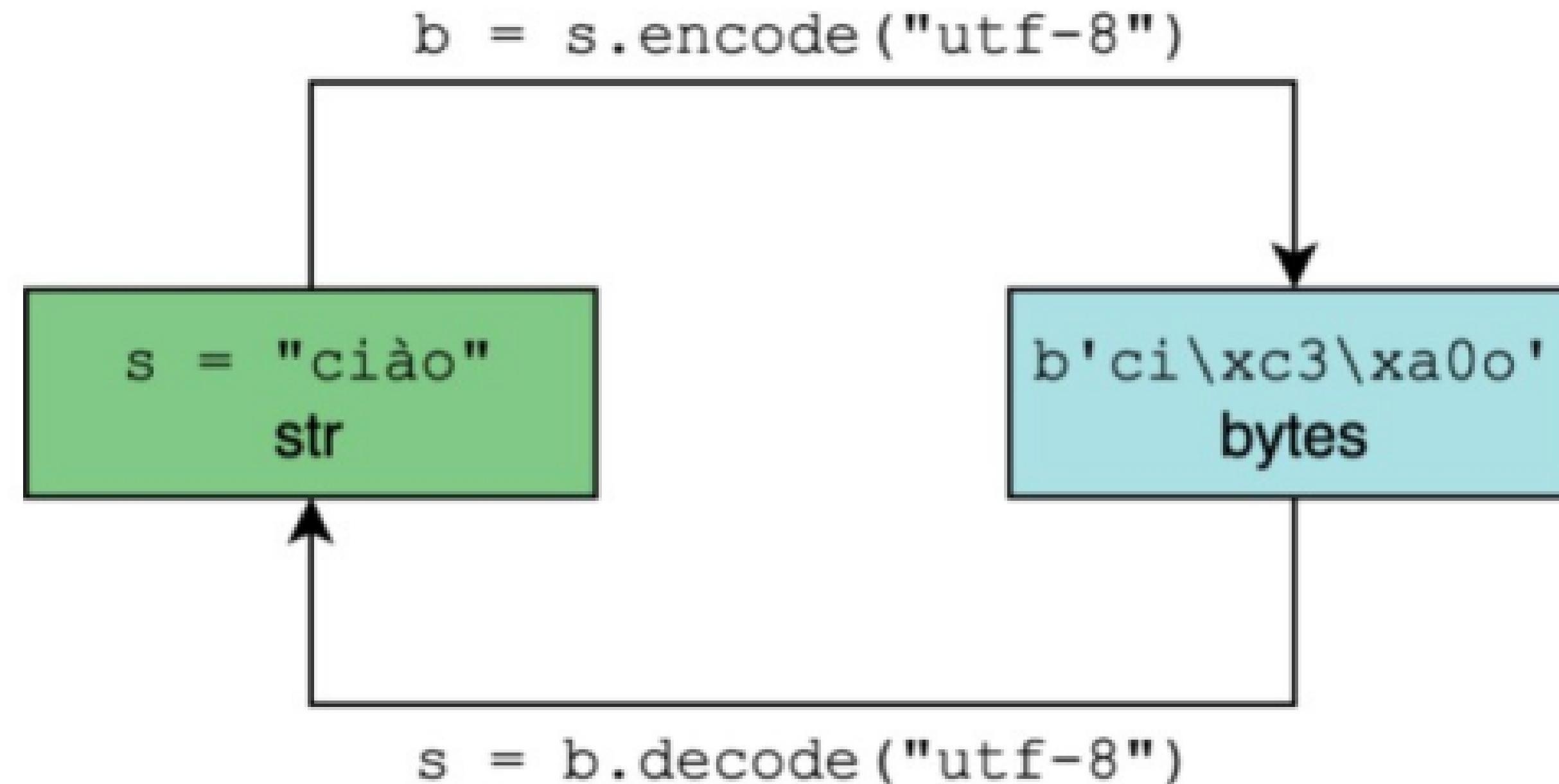
Код	Символ																
32		61	=	90	z	119	w	148	"	177	±	206	о	235	л		
33	!	62	>	91	[120	х	149	•	178	!	207	п	236	м		
34	"	63	?	92	\	121	у	150	-	179	і	208	р	237	н		
35	#	64	@	93]	122	з	151	-	180	г	209	с	238	о		
36	\$	65	А	94	^	123	{	152		181	μ	210	т	239	п		
37	%	66	В	95	-	124		153	™	182	ї	211	ү	240	р		
38	&	67	С	96	:	125	}	154	љ	183	.	212	ф	241	с		
39	'	68	Д	97	а	126	~	155	›	184	ё	213	х	242	т		
40	(69	Е	98	б	127	Ї	156	њ	185	№	214	ц	243	ү		
41)	70	Ф	99	с	128	ъ	157	ќ	186	е	215	ч	244	ф		
42	*	71	Г	100	д	129	ѓ	158	Ћ	187	»	216	ш	245	х		
43	+	72	Н	101	е	130	,	159	Ѱ	188	ј	217	Щ	246	ц		
44	,	73	І	102	ƒ	131	ѓ	160		189	Ѕ	218	ъ	247	ч		
45	-	74	Ј	103	ѕ	132	„	161	ӻ	190	ѕ	219	ы	248	ш		
46	.	75	Ќ	104	һ	133	...	162	ӻ	191	ї	220	ь	249	щ		
47	/	76	Љ	105	и	134	†	163	Ј	192	А	221	Э	250	ъ		
48	0	77	М	106	ј	135	‡	164	ќ	193	Б	222	Ю	251	ы		
49	1	78	Ң	107	к	136	€	165	Ѓ	194	В	223	Я	252	в		
50	2	79	Ӯ	108	l	137	‰	166	፣	195	Г	224	а	253	э		
51	3	80	҆	109	м	138	љ	167	Ӯ	196	Д	225	б	254	ю		
52	4	81	҇	110	п	139	‘	168	Ҽ	197	Ե	226	в	255	я		
53	5	82	҈	111	օ	140	њ	169	Ը	198	Ж	227	г				
54	6	83	҉	112	ր	141	ќ	170	Ը	199	З	228	д				
55	7	84	Ҋ	113	զ	142	Ћ	171	«	200	И	229	е				
56	8	85	ҋ	114	Ր	143	Џ	172	-	201	Й	230	ж				
57	9	86	Ҍ	115	ս	144	Ծ	173	-	202	Կ	231	з				
58	:	87	ҍ	116	տ	145	՚	174	•	203	Լ	232	и				
59	;	88	Ҏ	117	ս	146	՚	175	Ӯ	204	Մ	233	й				
60	<	89	ҏ	118	Վ	147	՞	176	•	205	Հ	234	կ				

Unicode

Что это такое?

- стандарт кодирования символов, включающий в себя значи почти всех письменных языков мира
- В настоящее время является преобладающим в интернете
- наибольшую популярность получила **UTF-8** - кодировка с переменным числом байт, она позволяет кодировать символы так, что наиболее распространенные символы занимают 1-2 байта, и только редко встречающиеся символы могут занимать по 4 байта

decode / encode



Режимы работы с файлами

Режим	Обозначение
'r'	открытие на чтение (является значением по умолчанию).
'w'	открытие на запись, содержимое файла удаляется, если файла не существует, создается новый.
'x'	открытие на запись, если файла не существует, иначе исключение.
'a'	открытие на дозапись, информация добавляется в конец файла.
'b'	открытие в двоичном режиме.
't'	открытие в текстовом режиме (является значением по умолчанию).
'+'	открытие на чтение и запись

Чтение файла

```
# чтение
f = open("txt_example.txt")
data = f.readlines()
f.close()
```

```
data
```

```
['1 line1\n', '2 line2\n', '3 line3']
```

Предпочтительный способ чтения файла

что такое `with`

```
# можно заменить на try:...except:..., но лучше использовать with
with open("new_txt_example.txt") as f:
    print(f.read())
```

Message

Работа с файлами

Способ №1

```
f = open('example.txt','r')  
# работа с файлом  
f.close()
```

Способ №2

```
f = open('example.txt','r')  
try:  
    # работа с файлом  
finally:  
    f.close()
```

Способ №3

```
with open('example.txt') as f:  
    # работа с файлом
```

Чтение и запись файлов в Python

Функция `read()`

```
f = open('example.txt','r')  
f.read(7) # чтение 7  
символов из example.txt  
f.read() # чтение всего  
файла
```

Функция `readline()`

```
f = open('test.txt','r')  
f.readline() # прочитать  
первую строку  
f.readlines() # прочитать все  
строки
```

Функция `write()`

```
f = open('xyz.txt','w') #  
открытие в режиме записи  
f.write('Hello \n World') #  
запись Hello World в файл  
f.close() # закрытие файла
```

Виды файлов - csv

1 CSV

Имя	Профессия	Год рождения
Виктор	Токарь	1995
Сергей	Сварщик	1983

Файл **CSV** – это особый вид файла, который позволяет структурировать большие объемы данных.

По сути, он является обычным текстовым файлом, однако каждый новый элемент отделен от предыдущего запятой или другим разделителем. Обычно каждая запись начинается с новой строки.

Данные CSV можно легко экспортить в электронные таблицы или базы данных.

Виды файлов - csv

2 CSV

Для того, чтобы прочитать данные из файла нужно создать объект reader:

```
1 reader_object = csv.reader(file, delimiter = ",")
```

reader имеет метод `__next__()`, то есть является итерируемым объектом.

Виды файлов - csv

3 CSV

```
import csv
with open("classmates.csv", encoding='utf-8') as r_file:
    # Создаем объект reader, указываем символ-разделитель ","
    file_reader = csv.reader(r_file, delimiter = ",")  
    # Счетчик для подсчета количества строк и вывода заголовков столбцов
    count = 0
    # Считывание данных из CSV файла
    for row in file_reader:
        if count == 0:
            # Вывод строки, содержащей заголовки для столбцов
            print(f'Файл содержит столбцы: {", ".join(row)}')
        else:
            # Вывод строк
            print(f'{row[0]} - {row[1]} и он родился в {row[2]} году.')
            count += 1
    print(f'Всего в файле {count} строк.')
```

Виды файлов - json

1 JSON

JSON - текстовый формат обмена данными, основанный на JavaScript. Но при этом формат независим от JS и может использоваться в любом языке программирования.

2 JSON

Для работы с json, нужно импортировать библиотеку json

```
1 import json
```

Виды файлов - json

3 JSON

Используя контекстный менеджер Python, вы можете создать файл под названием `data_file.json` и открыть его в режиме `write` (файлы JSON имеют расширение `.json`).

```
1 with open("data_file.json", "w") as write_file:  
2     json.dump(data, write_file)
```

3

домашнее задание

Домашнее задание (часть 1)

1. Дописать функцию так, чтобы она возвращала None в случае если ключа нет, а не генерировала исключение (для реализации используем исключения)

```
def func(dict_, key):  
    return dict_[key]
```

Домашнее задание (часть 2)

2. Ввести с клавиатуры 4 строки и сохранить их в 4 разные переменные. Создать файл и записать в него первые 2 строки и закрыть файл. Затем открыть файл на редактирование и дозаписать оставшиеся 2 строки. В итогом файле должны быть 4 строки, каждая из которых должна начинаться с новой строки.
3. Создать словарь в качестве ключа которого будет 6-ти значное число (id), а в качестве значений кортеж состоящий из 2-х элементов – имя(str), возраст(int). Сделать около 5-6 элементов словаря. Записать данный словарь на диск в json-файл.
4. Прочитать сохранённый json-файл и записать данные на диск в csv-файл, первой строкой которого озаглавив каждый столбец и добавив новый столбец “телефон”.
5. *Прочитать сохранённый csv-файл и сохранить данные в excel-файл, кроме возраста - столбец с этими данными не нужен. Таблица должна выглядеть, как на примере:

	A	B	C	D
1		person 1	Person 2	Person 3
2	id	111111	222043	679432
3	name	Sam	Adam	Jasy
4	phone	097-32-88	090-46-41	094-66-22

Спасибо за внимание!