IT CAREER HUB

Python

Знакомство со строками

Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно

- Камера должна быть включена на протяжении всего занятия
- В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- Вести себя уважительно и этично по отношению к остальным участникам занятия
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение

- Цикл
- Оператор break
- Оператор continue
- Бесконечный цикл
- Конструкция while/else
- Модуль random

План занятия

- Неизменяемость строк
- Кодировка
- Функции ord, chr
- Сравнение строк
- Функция len
- Индексация строк
- Срезы строк
- Оператор принадлежности in

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

ICH IT CAREER HUB

Неизменяемость строк

Полезно знать

ICH
IT CAREER HUB

Неизменяемость строк

В Python строки являются неизменяемыми (`immutable`) объектами. Это означает, что после создания строки её содержимое нельзя изменить напрямую.

Пример неизменяемости строки

```
s = "Hello"
print(s + " world!")
print(s)
```

Важно

Если нужно "изменить" строку, это делается через создание новой строки и присваивания её старой переменной

"Изменение" содержимого строки

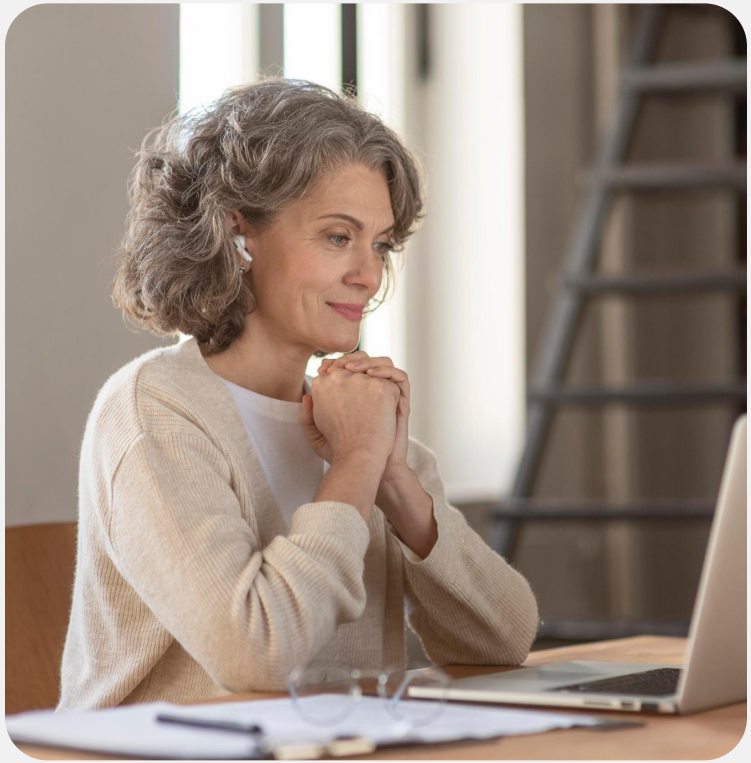
```
s = "Hello"
s = s + " world!"
print(s)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

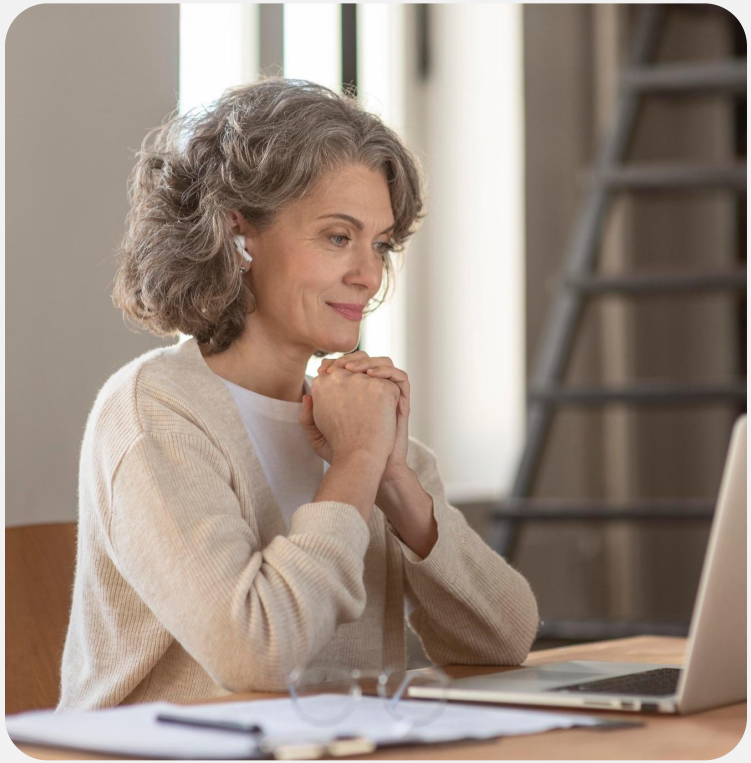


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
s = "Python"
s += " is fun"
print(s)
```

a. Python
b. Python is fun
c. Ошибка
d. Python is



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
s = "Python"
s += " is fun"
print(s)
```

a. Python
b. Python is fun
c. Ошибка
d. Python is

IT CAREER HUB

Кодировка

Полезно знать

ICH
IT CAREER HUB

Кодировка

Это способ преобразования символов в числа, которые компьютер может хранить и обрабатывать.

Полезно знать

ICH
IT CAREER HUB

Кодировка ASCII

Это одна из первых и простейших кодировок, разработанная для представления текстовых символов в компьютерах.

Характеристики кодировки ASCII

7 бит

ASCII использует 7 бит для кодирования символов.

Это позволяет закодировать 128 символов (2^7 = 128).

Символы представлены числами от 0 до 127.

Символы

0-31: Управляющие символы (например, символ новой строки, возврат каретки, сигнал окончания текста).

32-126: Печатные символы, включая буквы, цифры, пунктуацию и другие символы.

Простота и совместимость

ASCII был создан для работы с английским алфавитом, цифрами и основными символами.

Он широко используется в компьютерах, сетевых системах и других устройствах, обеспечивая базовый уровень кодирования для текстовых данных.

Таблица символов ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Расширения ASCII

Оригинальный ASCII использует 7 бит для представления 128 символов.

Расширенные версии используют 8 бит, позволяя представлять 256 символов (0–255)

Расширения ASCII

Расширенные символы (128–255)

- Символы национальных алфавитов (например, буквы с диакритическими знаками: é, ü, ç).
- Графические символы и псевдографика (например, рамки, линии, блоки).
- Дополнительные спецсимволы (валютные знаки, математические символы и т. д.).

Примеры стандартов

- **ISO 8859-1 (Latin-1):** Для западноевропейских языков. Поддерживает символы, такие как ñ, ß, ø.
- **Windows-1251:** Для славянских языков, включая кириллицу.
- **KOI8-R:** Для русского языка, популярная в UNIX-системах.
- **MacRoman:** Для компьютеров Apple.

Полезно знать

Unicode

Это глобальный стандарт, который присваивает каждому символу уникальный числовой код, называемый кодовой точкой. Он охватывает символы всех языков мира, математические символы, эмодзи и многое другое.

Категории символов в Unicode

- Алфавиты различных языков
- Математические и технические символы
- Символы для форматирования текста
- Эмодзи
- Диакритические знаки и другие спецсимволы

Полезно знать

ICH
IT CAREER HUB

UTF-8

Это одна из самых популярных и широко используемых кодировок для представления символов в Unicode. Она преобразует символы Unicode в последовательности байтов переменной длины, где каждый символ может занимать от 1 до 4 байт.

Как работает кодировка UTF-8

- **1 байт:** Символы ASCII (от 0 до 127), например, латинские буквы, цифры и некоторые знаки пунктуации.
- **2 байта:** Символы с кодами от 128 до 2047, например, символы латиницы с диакритическими знаками (акценты), кириллические буквы и греческий алфавит.
- **3 байта:** Символы с кодами от 2048 до 65535, включающие китайские иероглифы, японские кандзи и хангыль.
- **4 байта:** Символы с кодами выше 65536, такие как редкие символы и эмодзи.

Преимущества UTF-8:

Эффективное использование памяти

Для текстов на латинице (английский, испанский и т.д.) UTF-8 экономит память, используя только 1 байт на символ.

Для текстов на других языках, таких как русский или китайский, UTF-8 остаётся эффективной, занимая меньше памяти по сравнению с UTF-16 или UTF-32.

Гибкость

UTF-8 может кодировать символы любой сложности: от простых букв латинского алфавита до редких символов и эмодзи.

Международный стандарт

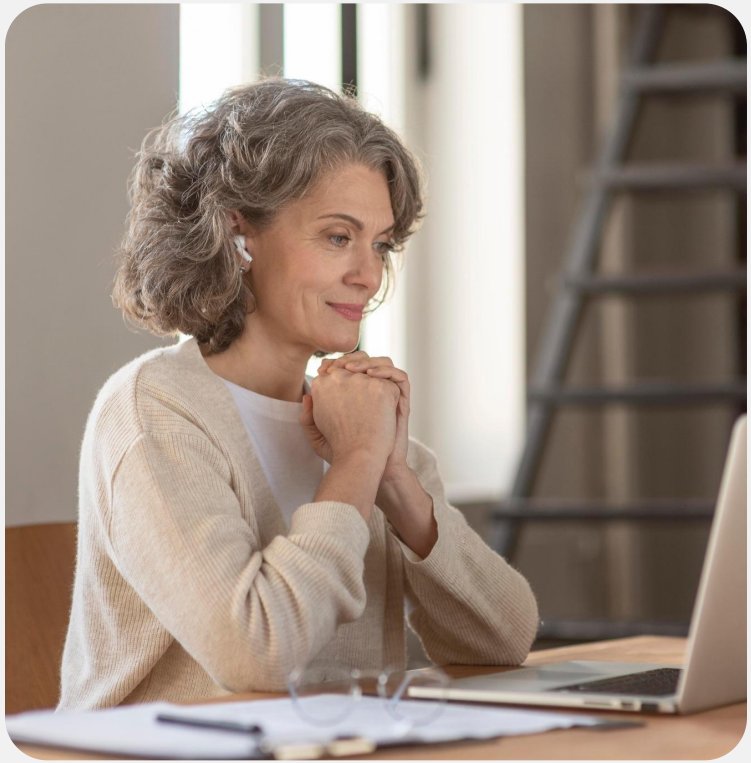
UTF-8 стал основным стандартом кодирования текстов в Интернете. Большинство веб-страниц и веб-приложений используют UTF-8.

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

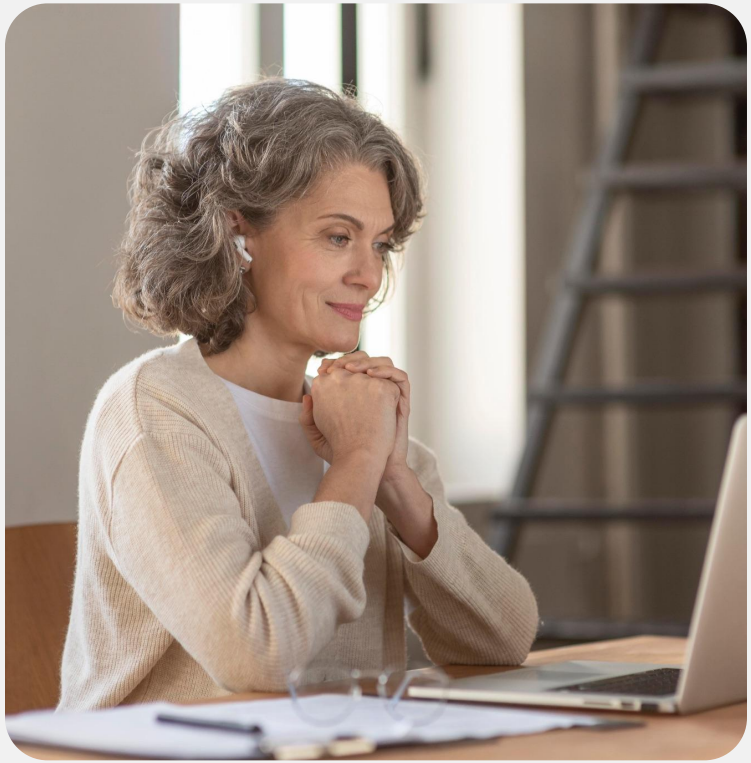
ЗАДАНИЕ



Выберите правильный вариант ответа

В чем отличие между кодировками ASCII и UTF-8?

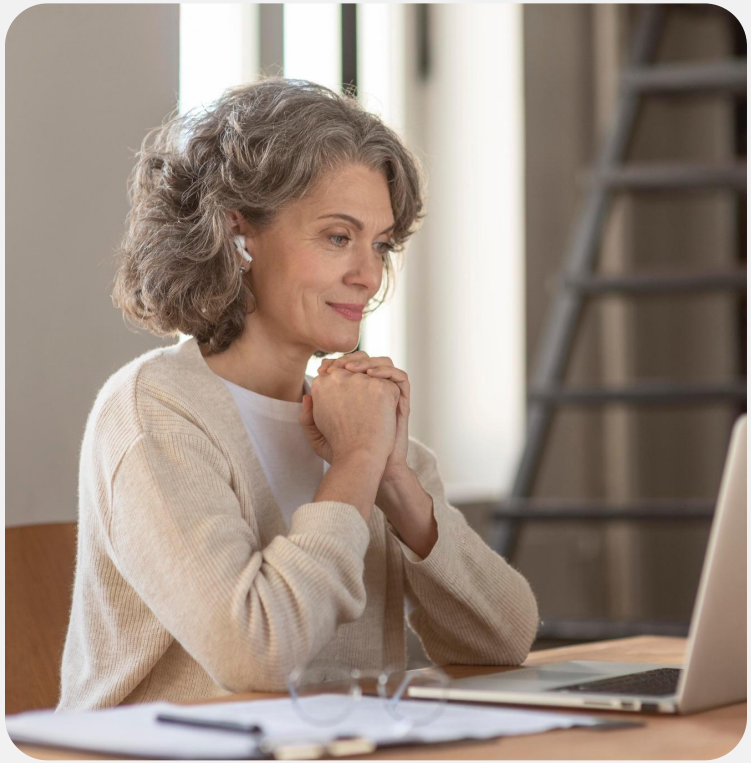
a. UTF-8 поддерживает больше символов, чем ASCII.
b. ASCII поддерживает больше символов, чем UTF-8.
c. В ASCII и UTF-8 одинаковое количество символов.
d. UTF-8 использует 7 бит для кодирования символов.



Выберите правильный вариант ответа

В чем отличие между кодировками ASCII и UTF-8?

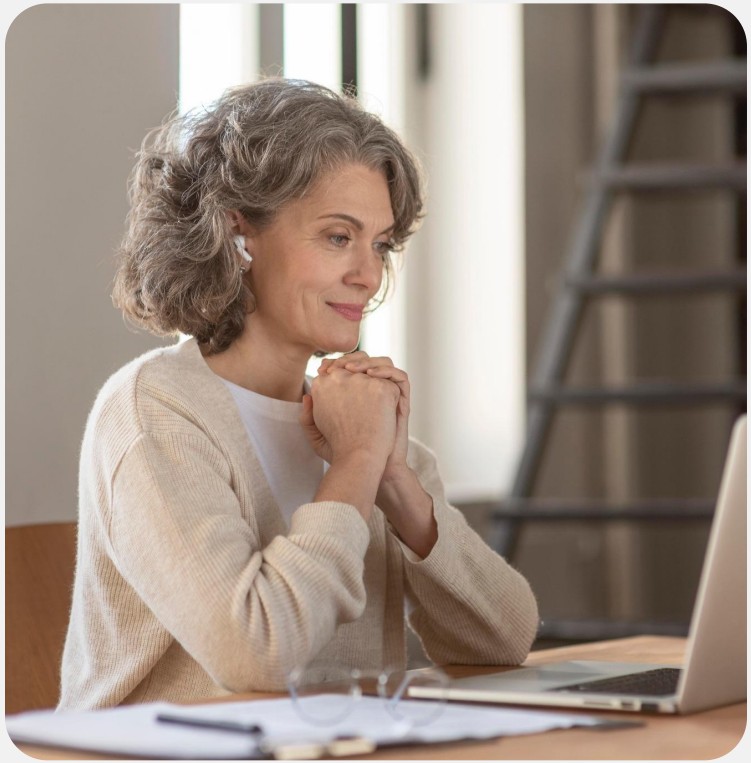
a. **UTF-8 поддерживает больше символов, чем ASCII.**
b. ASCII поддерживает больше символов, чем UTF-8.
c. В ASCII и UTF-8 одинаковое количество символов.
d. UTF-8 использует 7 бит для кодирования символов.



Выберите правильный вариант ответа

Какой из следующих диапазонов символов поддерживается кодировкой ASCII?

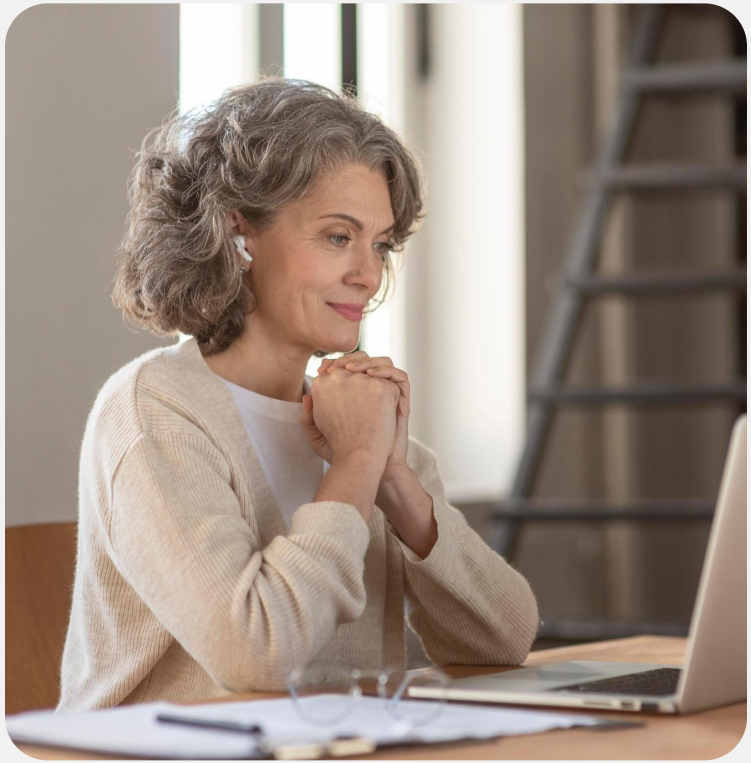
a. Символы от 0 до 127
b. Символы от 0 до 255
c. Символы от 0 до 1024
d. Символы от 0 до 65535



Выберите правильный вариант ответа

Какой из следующих диапазонов символов поддерживается кодировкой ASCII?

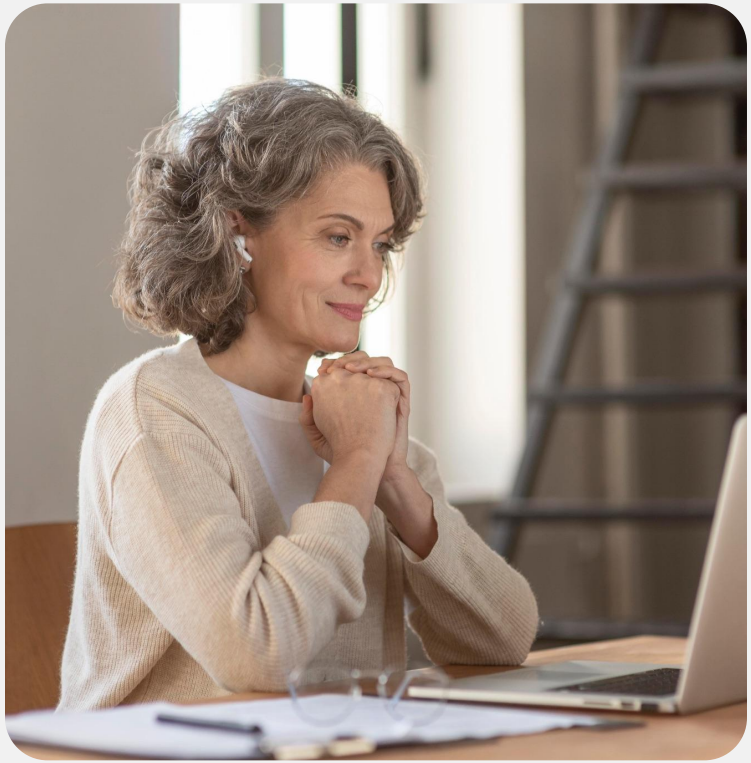
a. **Символы от 0 до 127**
b. Символы от 0 до 255
c. Символы от 0 до 1024
d. Символы от 0 до 65535



Выберите правильный вариант ответа

Какой из вариантов неверен относительно Unicode?

a. Unicode поддерживает символы всех языков мира.
b. В Unicode содержится больше 143 000 символов.
c. Unicode использует 1 бит для кодирования всех символов.
d. Unicode охватывает символы эмодзи и технические знаки.



Выберите правильный вариант ответа

Какой из вариантов неверен относительно Unicode?

a. Unicode поддерживает символы всех языков мира.
b. В Unicode содержится больше 143 000 символов.
c. Unicode использует 1 бит для кодирования всех символов.
d. Unicode охватывает символы эмодзи и технические знаки.

IT CAREER HUB

Метод

Полезно знать

ICH
IT CAREER HUB

Метод

Это функция, которая принадлежит определённому объекту или классу и может взаимодействовать с его данными

Связь метода с объектом

Объяснение

Методы вызываются через объекты (экземпляры классов) и могут работать с данными, принадлежащими этим объектам

Пример

Строка в Python является объектом класса `str`, и у неё есть методы для работы с текстом

Вызов метода

Синтаксис

```
object_name.method_name
```

Пример

```
text = "hello"
print(text.encode('utf-8'))
# Кодирует текст, полученный из
# переменной text в 'utf-8'
```

Аргументы методов

Методы могут принимать аргументы, которые помогают управлять их поведением

IT CAREER HUB

Методы encode и decode

Полезно знать

ICH
IT CAREER HUB

encode()

Этот метод преобразует строку в байты, используя указанную кодировку (например, UTF-8). Это полезно при передаче данных по сети или записи в файл.

Полезно знать

IT CAREER HUB

decode()

Этот метод преобразует байты обратно в строку, интерпретируя их как символы на основе кодировки

Пример кодирования строки в UTF-8

```
text = "Python"
# Метод .encode() кодирует строку в байты с использованием UTF-8
encoded_text = text.encode('utf-8')
print(encoded_text)


# Оригинальная строка
text = "Привет"

# Кодирование строки в байты с использованием UTF-8
encoded_text = text.encode('utf-8')
print(encoded_text)
```

Декодирование строки обратно в текст

```
# Декодирование байтов обратно в строку
decoded_text = encoded_text.decode('utf-8')
print(decoded_text)
```

Где нужны кодировки

- Работа с файлами
- Работа с сетевыми данными
- Международные приложения

IT CAREER HUB

Функции ord, chr

Полезно знать

IT CAREER HUB

ord() и chr()

Это встроенные функции Python, которые работают с символами и их кодировкой.

Полезно знать

Функция ord()

Принимает символ и возвращает его код Unicode.

Пример

```
print(ord('A'))

print(ord('a'))

print(ord(' '))
```

Полезно знать

IT CAREER HUB

Функция chr()

Принимает число, представляющее код Unicode, и возвращает соответствующий символ.

Пример

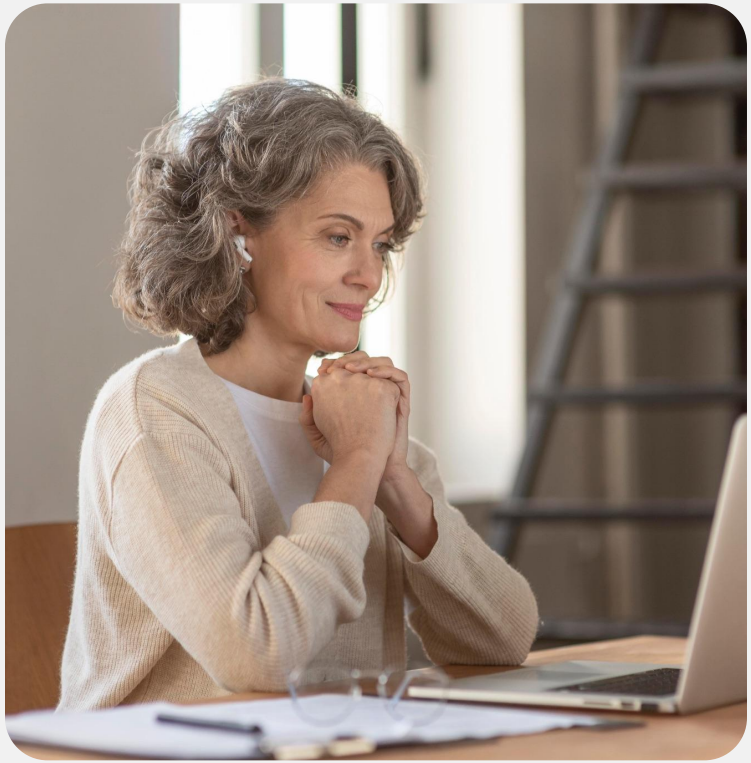
```
print(chr(65))

print(chr(97))

print(chr(32))
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

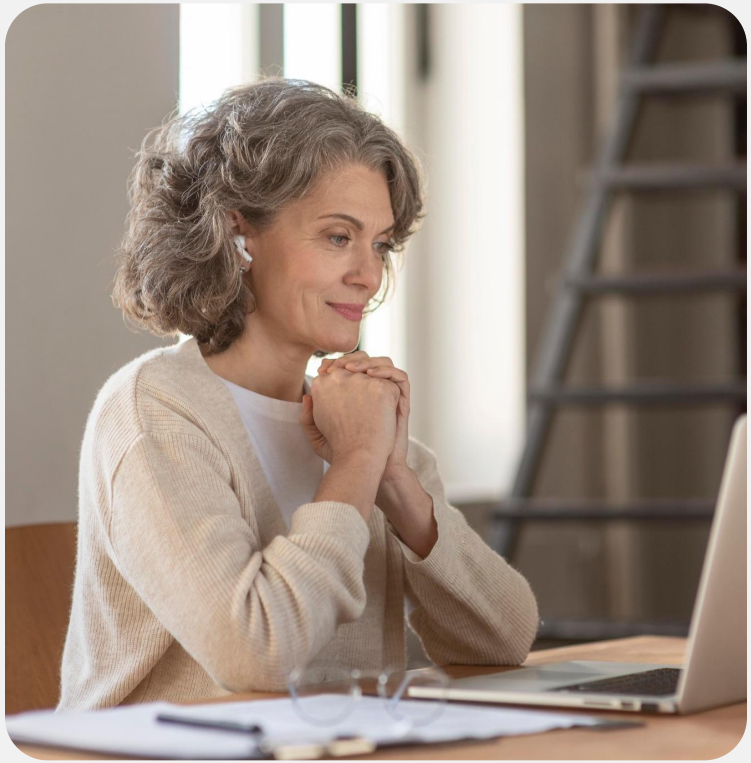
ЗАДАНИЕ



Выберите правильный вариант ответа

Что делает функция ord() в Python?

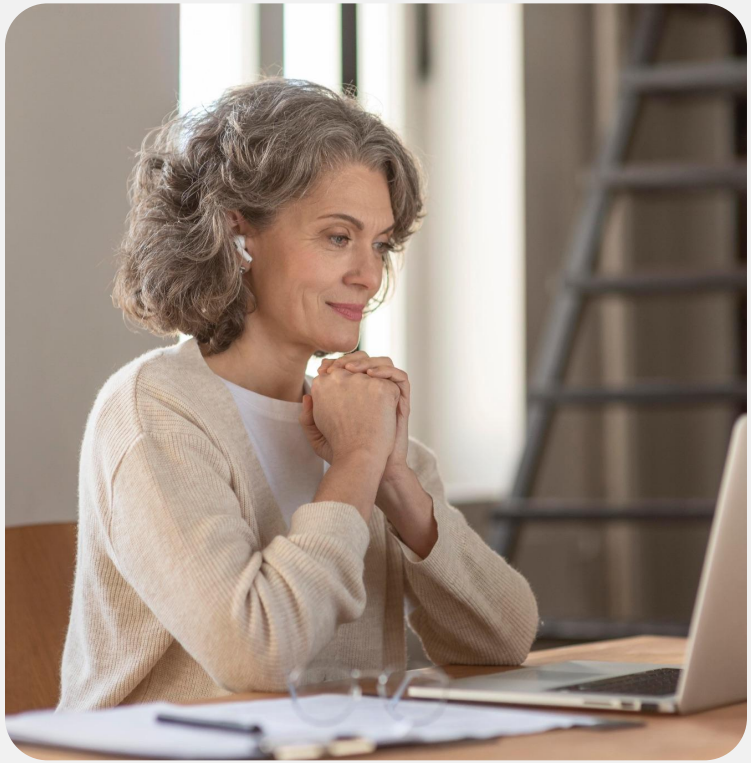
a. Преобразует строку в байты
b. Возвращает числовой код символа Unicode
c. Преобразует число в строку
d. Возвращает кодировку символа



Выберите правильный вариант ответа

Что делает функция ord() в Python?

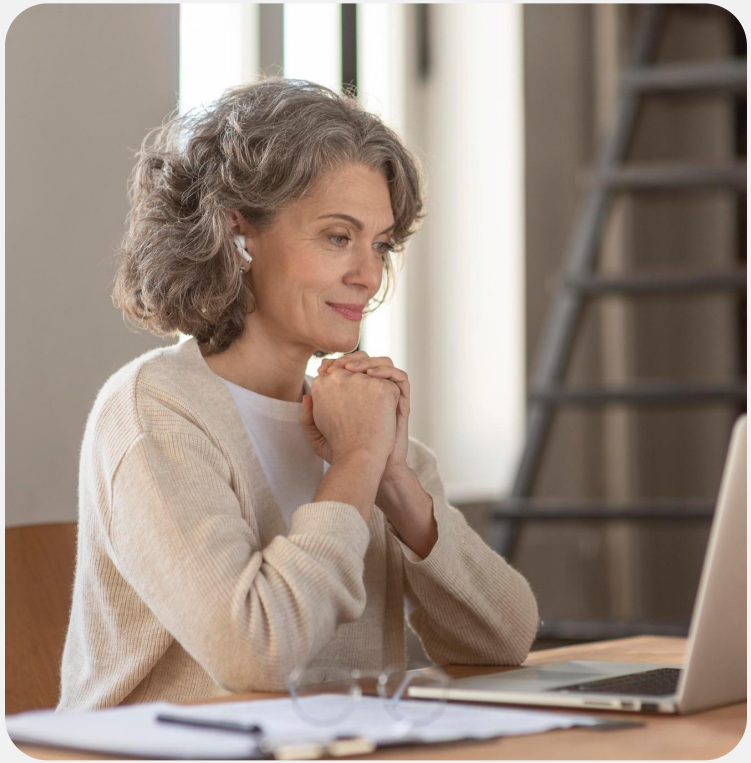
a. Преобразует строку в байты
b. Возвращает числовой код символа Unicode
c. Преобразует число в строку
d. Возвращает кодировку символа



Выберите правильный вариант ответа

Какой метод используется для преобразования строки в байты с указанием кодировки?

a. decode()
b. ord()
c. encode()
d. chr()



Выберите правильный вариант ответа

Какой метод используется для преобразования строки в байты с указанием кодировки?

a. decode()
b. ord()
c. encode()
d. chr()

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Сравнение строк

Полезно знать

ICH
IT CAREER HUB

Сравнение строк

В Python строки можно сравнивать с помощью операторов сравнения, таких как ==, !=, <, >, <= и >=.
Сравнение строк происходит на основе **лексикографического порядка**, что означает, что строки сравниваются по символам с использованием их кода в Unicode

Оператор ==

Действие

- Проверяет равенство двух строк.

Код

```
print("apple" == "apple")

print("apple" == "banana")
```

Оператор !=

Действие

- Проверяет неравенство двух строк.

Код

```
print("apple" != "ap ple")
```

Операторы <, >, <=, >=

Действие

- Сравнивают строки лексикографически.
- Строки сравниваются символ за символом на основе их Unicode кодов.

Код

```
print("apple" < "banana")
# True('a' < 'b')
print("Apple" < "banana")
# True ('A' < 'b')
print("apple" > "Apple")
# True ('a' > 'A')
```

Как происходит сравнение

Действие

- **Лексикографический порядок:** Python сравнивает строки символ за символом.
- Как только найдено различие между двумя символами, сравнение завершается.

Код

```
print("abc" < "aca")

# True, так как 'b' < 'c'
```

Как происходит сравнение

Действие

- **Учёт регистра:** В Unicode строчные буквы (например, 'a') имеют больший код, чем заглавные (например, 'A').
- Поэтому заглавные буквы считаются "меньшими" по значению.

Код

```
print("apple" > "Apple")

# True, так как 'a' > 'A'
```

Как происходит сравнение

Действие

- **Сравнение строк разной длины:** Когда строки разной длины сравниваются, Python продолжает сравнение, пока не закончатся символы в одной из строк.
- Строка, у которой символы закончились первой, считается "меньшей".

Код

```
print("apple" < "app")

# False, так как "apple" не короче
```

Полезно знать

ICH
IT CAREER HUB

Функция len()

Эта функция используется для получения длины объектов, таких как строки и другие последовательности. Она возвращает количество элементов в объекте.

Использование функции len() для строк

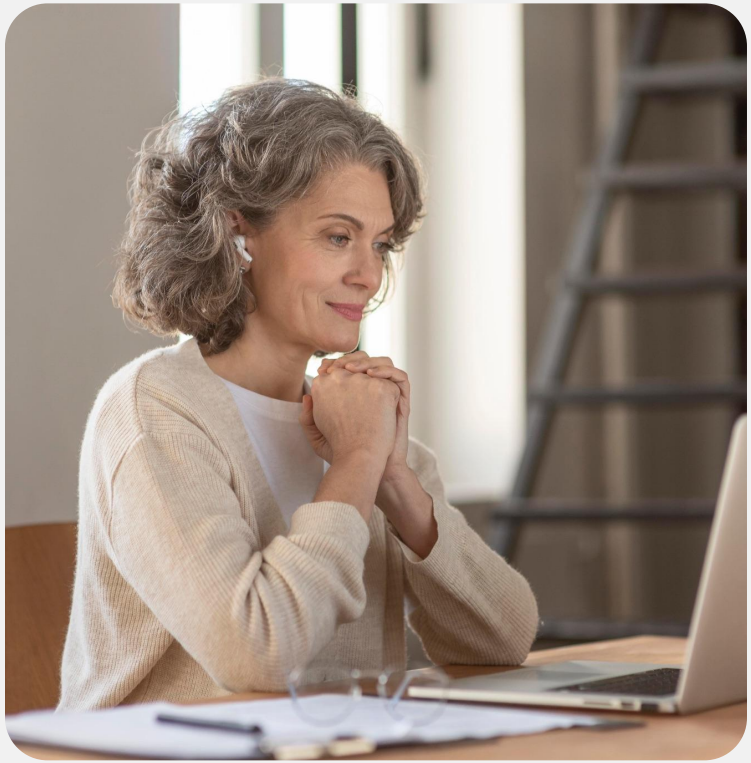
```
text = "Python"

length = len(text)

print(length)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

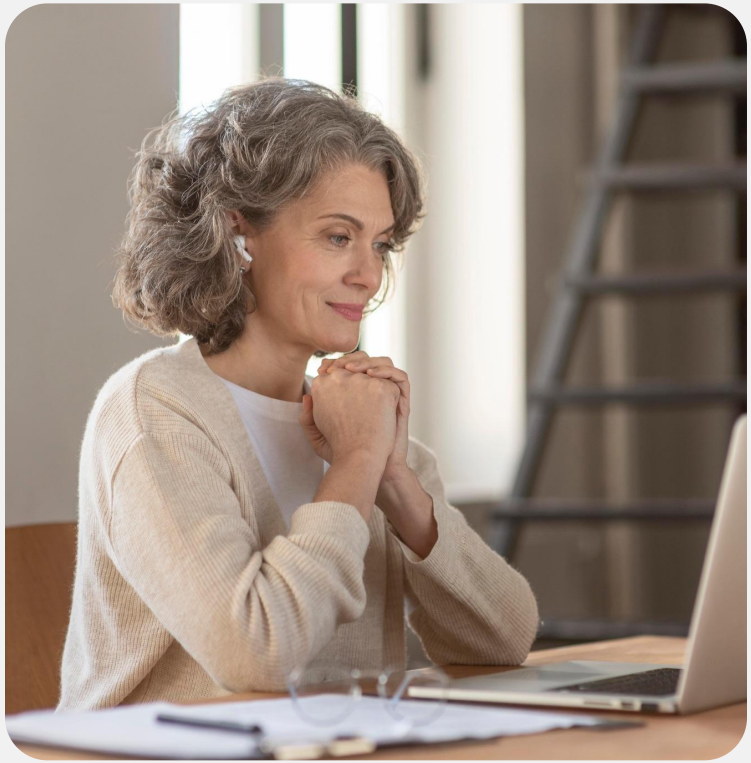


Выберите правильный вариант ответа

Что произойдет, если сравнить строки разной длины?

```
print("cat" > "catalog")
```

a. True
b. False
c. Ошибка
d. Невозможно сравнить строки

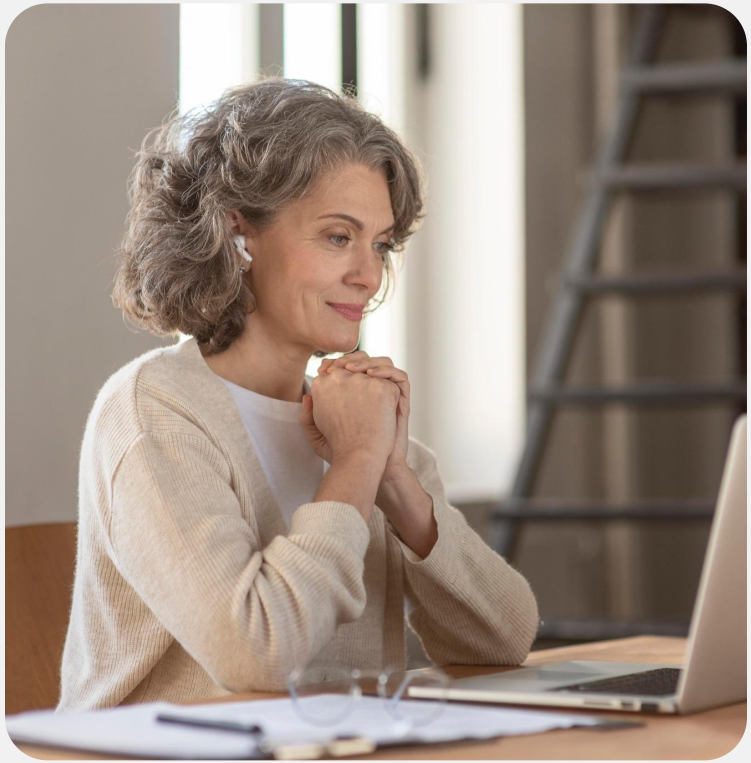


Выберите правильный вариант ответа

Что произойдет, если сравнить строки разной длины?

```
print("cat" > "catalog")
```

a. True
b. False
c. Ошибка
d. Невозможно сравнить строки

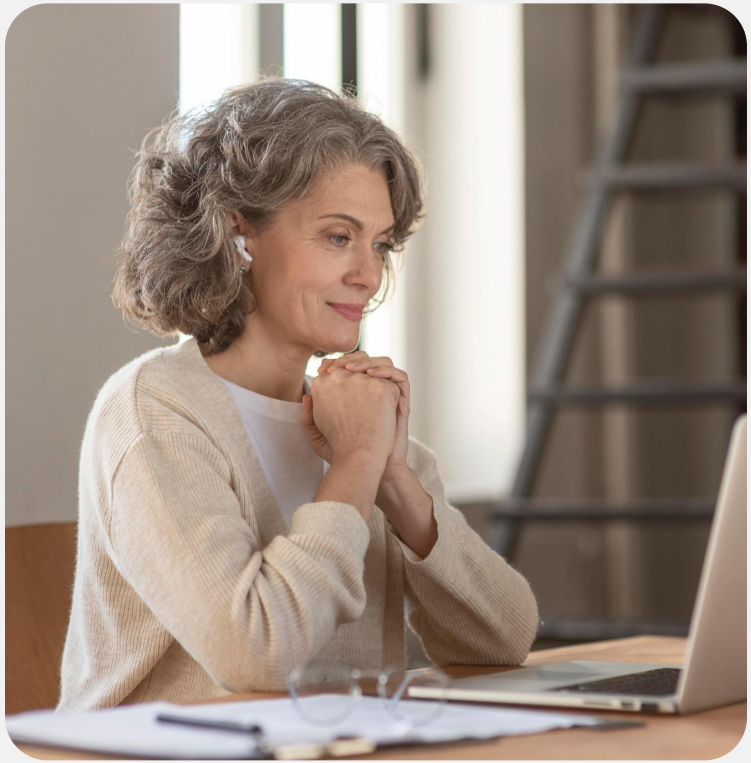


Выберите правильный вариант ответа

Какой результат будет выведен при сравнении строк с учётом регистра?

```
print("Zoo" < "zoo")
```

a. True
b. False
c. Ошибка
d. Невозможно сравнить строки

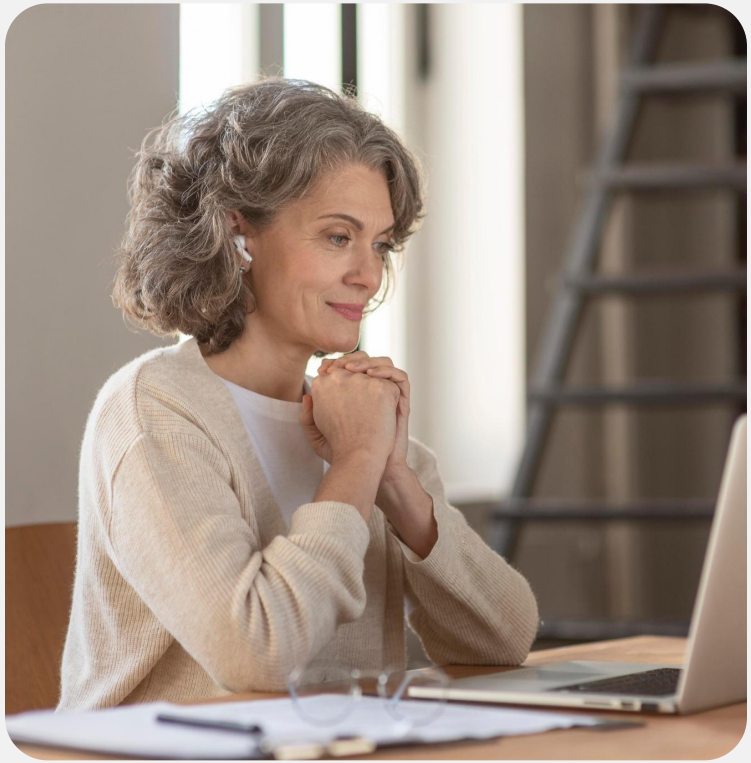


Выберите правильный вариант ответа

Какой результат будет выведен при сравнении строк с учётом регистра?

```
print("Zoo" < "zoo")
```

a. True
b. False
c. Ошибка
d. Невозможно сравнить строки

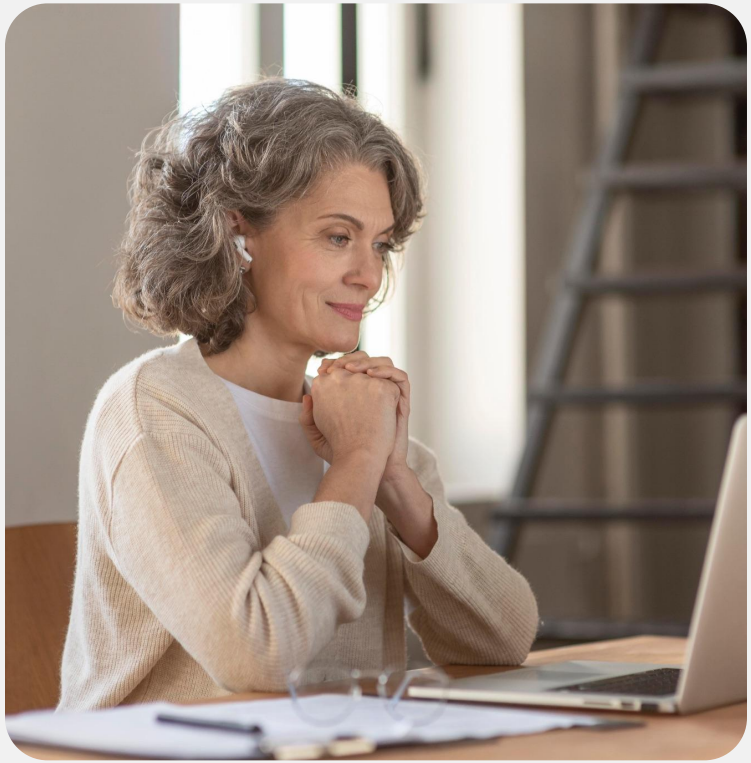


Выберите правильный вариант ответа

Какой результат будет выведен при сравнении строк с пробелом?

```
print("apple" != "ap ple")
```

a. True
b. False
c. Ошибка
d. Невозможно сравнить строки

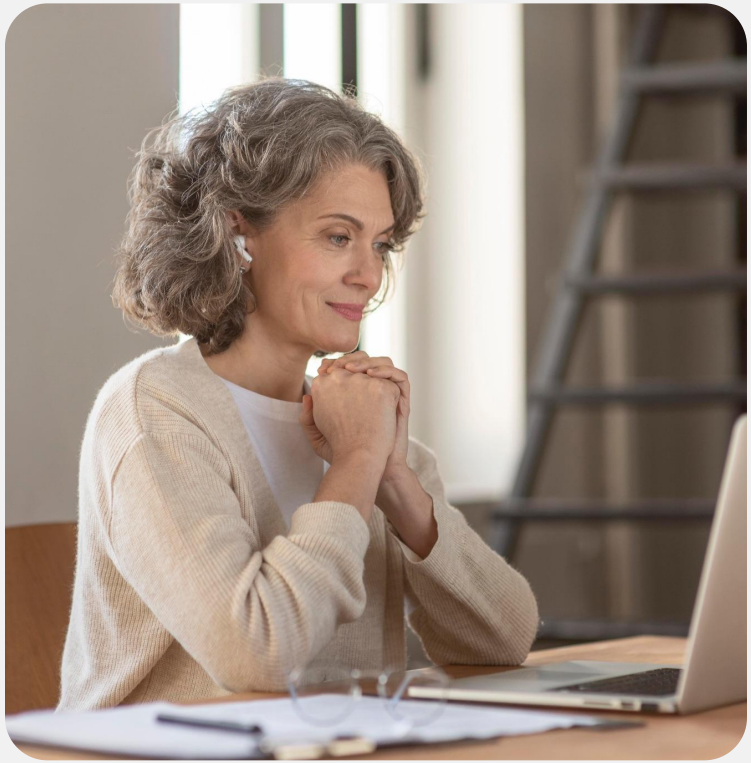


Выберите правильный вариант ответа

Какой результат будет выведен при сравнении строк с пробелом?

```
print("apple" != "ap ple")
```

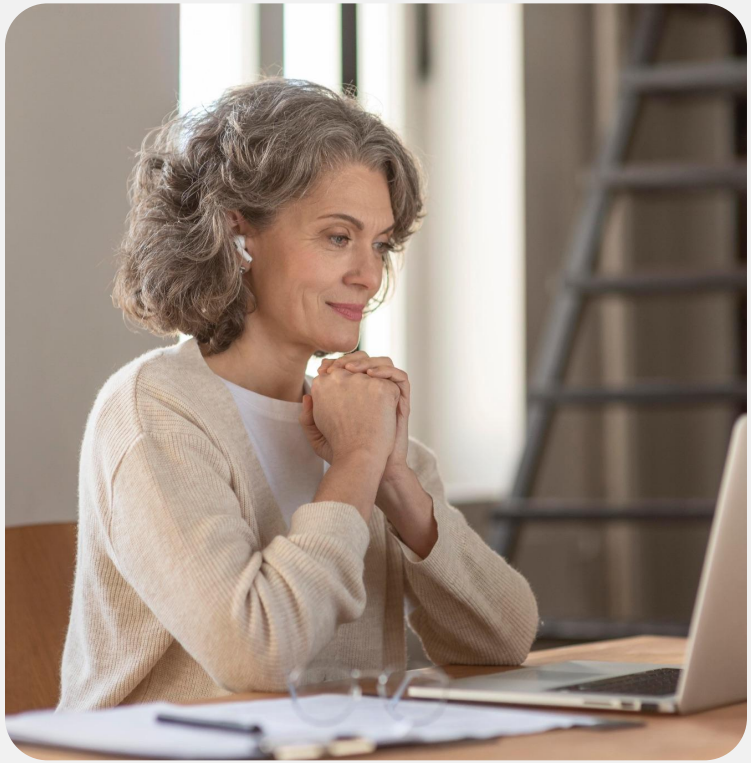
a. True
b. False
c. Ошибка
d. Невозможно сравнить строки



Выберите правильный вариант ответа

Какая функция возвращает длину строки?

a. length()
b. len()
c. size()
d. count()



Выберите правильный вариант ответа

Какая функция возвращает длину строки?

a. length()
b. len()
c. size()
d. count()

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Индексация строк

Полезно знать

ICH
IT CAREER HUB

Индексация

Это механизм обращения к отдельным элементам в последовательных структурах данных, таких как строки и другие. В Python индексация начинается с **0**

Пример индексации

```
Для строк:

text = "Python"

print(text[0])   # Вывод: 'P' (первый символ)

print(text[2])   # Вывод: 't' (третий символ)
```

Отрицательная индексация

В Python можно использовать отрицательные индексы для обращения к элементам с конца последовательности. Последний элемент имеет индекс **-1**, предпоследний — **-2** и так далее.

Пример: отрицательная индексация

```
text = "Python"

print(text[-1])  # Вывод: 'n' (последний символ)

print(text[-3])  # Вывод: 'h' (третий символ с конца)
```

Ошибка при обращении по несуществующему индексу

```
text = "hello"

print(text[10])  # Ошибка, так как в строке "hello" всего 5 символов

print(text[-6])  # Ошибка, так как в строке "hello" всего 5 символов
```

```
text = "hello"
print(text[10])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[1], line 2
      1 text = "hello"
----> 2 print(text[10])

IndexError: string index out of range
```

Как избежать ошибки

Действие

- **Проверять длину строки:** Прежде чем обращаться к символу по индексу, можно убедиться, что этот индекс находится в пределах допустимой длины строки.
- Для этого используется функция len(), которая возвращает длину строки.

Код

```
text = "hello"
index = 5

if index < len(text):
    print(text[index])

else:
    print("Индекс вне диапазона!")
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Срезы строк

Полезно знать

ICH
IT CAREER HUB

Срезы

Это способ получения подстрок из существующей строки, используя индексы.

Срезы

Пояснение

- **start** — индекс, с которого начинается срез (включительно).
- **end** — индекс, на котором срез заканчивается (не включительно).

Синтаксис срезов

```
строка[start:end]
```

Пример среза строки

```
text = "Python programming"

print(text[0:6])
```

Срезы

Пояснения

1. Если не указывать начальный индекс, срез начнётся с начала строки
2. Если не указывать конечный индекс, срез продлится до конца строки
3. Если не указывать оба индекса, срез создаст копию всей строки

Примеры

1.
```
text = "Python programming"
print(text[:6])
```

2.
```
text = "Python programming"
print(text[7:])
```

3.
```
text = "Python programming"
print(text[:])
```

Отрицательные индексы

Пояснения

Можно использовать отрицательные индексы для срезов **с конца строки**. Например, индекс **-1** соответствует **последнему символу** строки

Пример

```
text = "Python programming"
print(text[-11:])
```

Полезно знать

Порядок индексов

При указании индекса нужно учитывать что `end` должен быть больше чем `start`. При указании некорректных индексов результатом будет пустая строка

Пример

```
text = "Python programming"
print(text[-11:-4])  # -11 < -4
print(text[7:14])  # 7 < 14

text = "Python programming"
print(text[-4:-11])  # Пустая строка
print(text[14:7])  # Пустая строка
```

Срезы с шагом

Пояснения

- **start** — начальный индекс (включительно).
- **end** — конечный индекс (не включительно).
- **step** — шаг, который определяет, через сколько символов брать элементы.

Синтаксис

```
строка[start:end:step]
```

Примеры

```
# Положительный шаг
text = "Python programming"
print(text[0:12:2])  # Вывод: 'Pto rg'

# Отрицательный шаг (обратный порядок)
text = "Python programming"
print(text[12:6:-1])  # Вывод: 'argorp'
```

Срезы с отрицательным шагом

Пояснения

- При указании отрицательного шага нужно учитывать что теперь **end** должен быть меньше чем **start**, т.к. выбор символов происходит в обратном порядке
- При указании некорректных индексов результатом будет пустая строка

Примеры

```
text = "Python programming"
print(text[-4:-11:-1])  # -11 < -4
print(text[14:7:-1])  # 7 < 14
```

```
text = "Python programming"
print(text[-11:-4:-1])  # Пустая строка
print(text[7:14:-1])  # Пустая строка
```

Срез всей строки в обратном порядке

```python
text = "Python"

print(text[::-1])  # Вывод: 'nohtyP'
```

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Оператор принадлежности in

Полезно знать

Оператор in

Используется для проверки, содержит ли строка определённый элемент или подстроку.

Пример с использованием оператора in для строк

```
text = "Python programming"

print("Python" in text)

print("Hello" in text)
```

Полезно знать

ICH
IT CAREER HUB

Комбинация not и in

Используется для проверки, что элемент не содержится в строке или другой последовательности. `not` просто инвертирует результат выражения с оператором `in`.

Пример

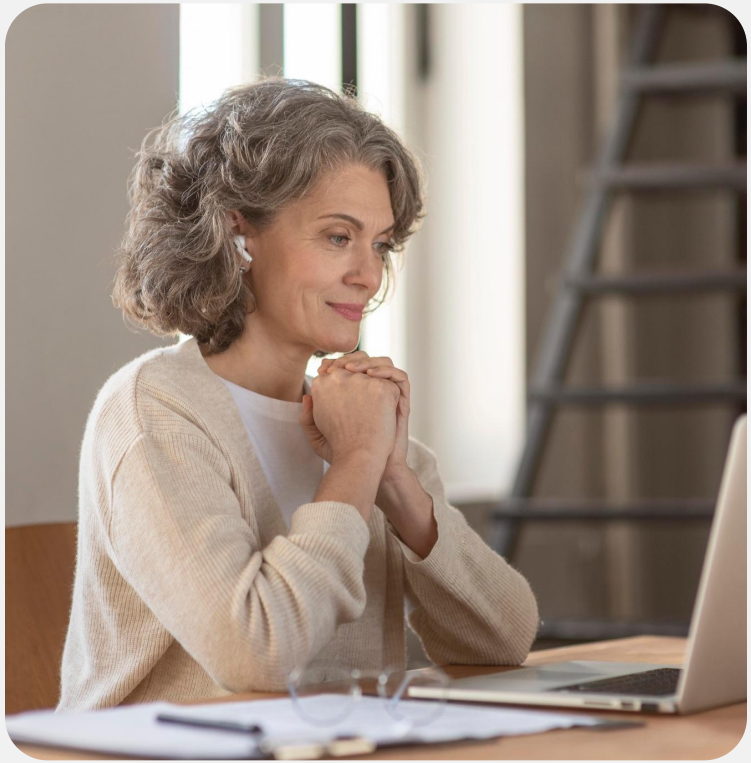
```
text = "Python programming"

print("Java" not in text)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

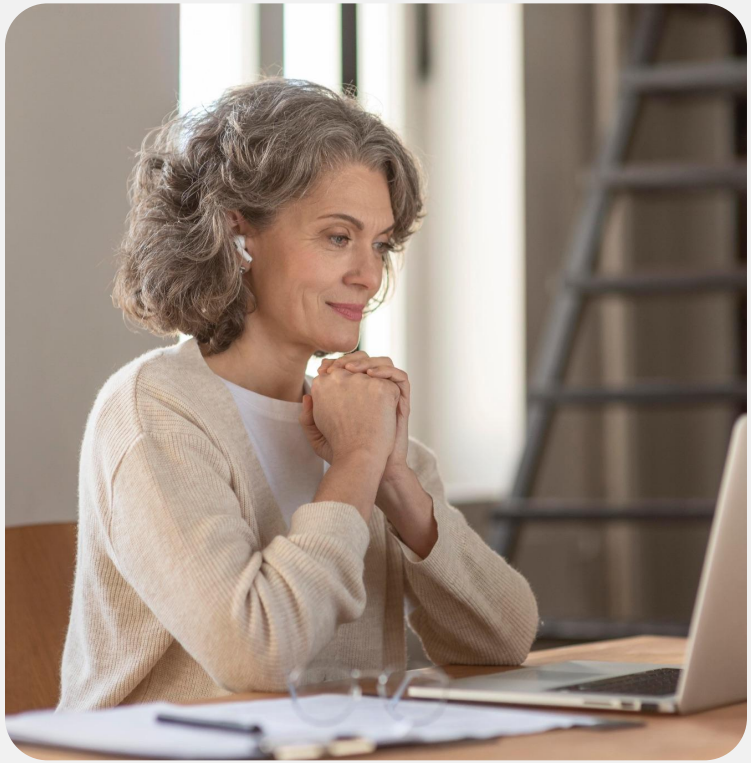


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[7])
```

a. H
b. o
c. P
d. Ошибка

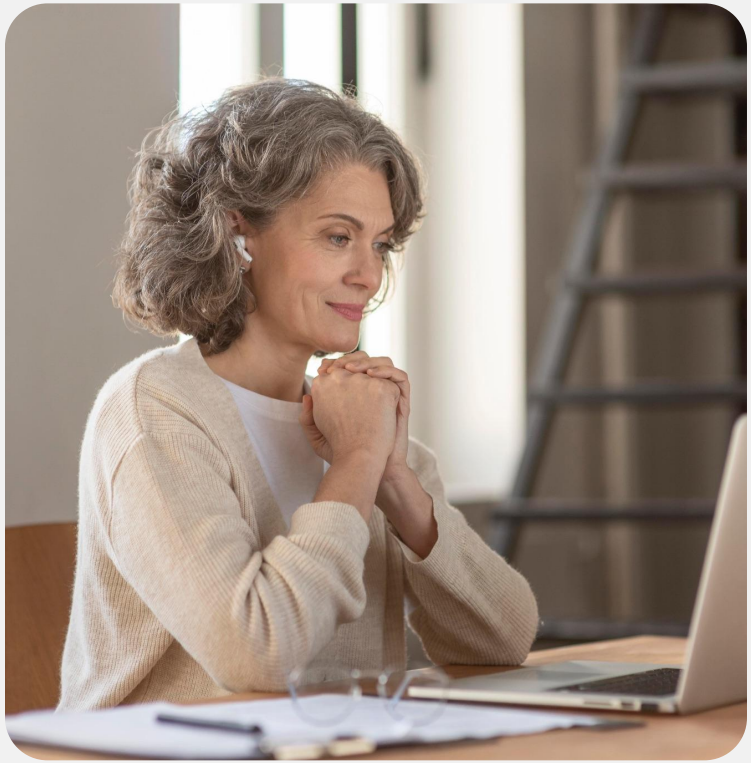


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[7])
```

a. H
b. o
c. P
d. Ошибка

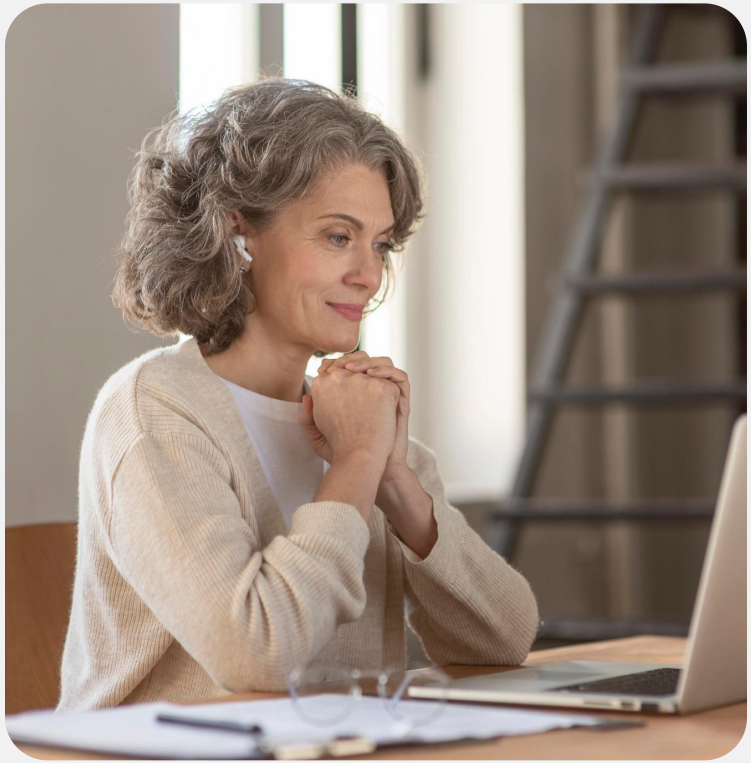


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[-3])
```

a. t
b. o
c. h
d. o

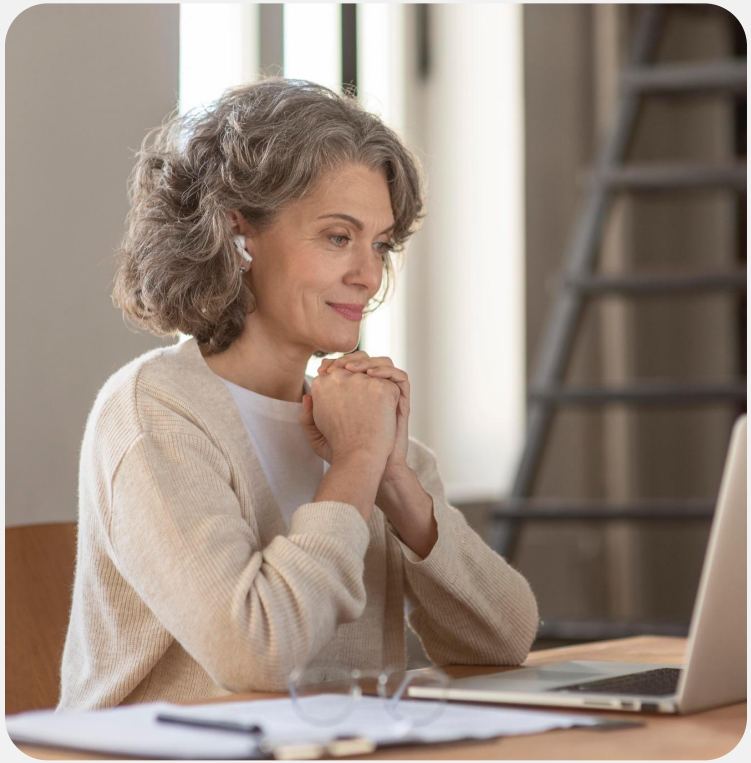


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[-3])
```

a. t
b. o
c. h
d. o

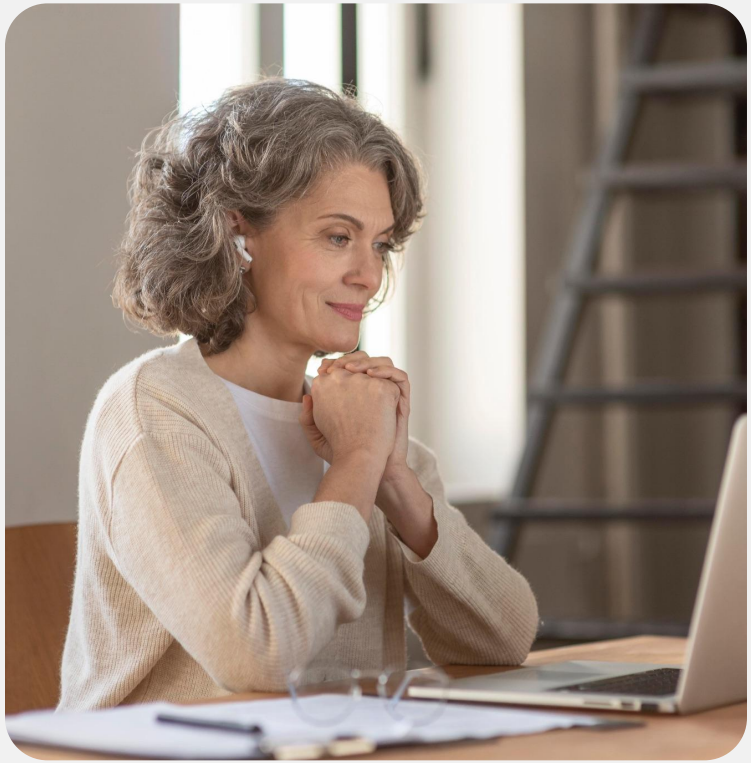


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[1:6])
```

a. Hello
b. ello,
c. ello
d. Ошибка

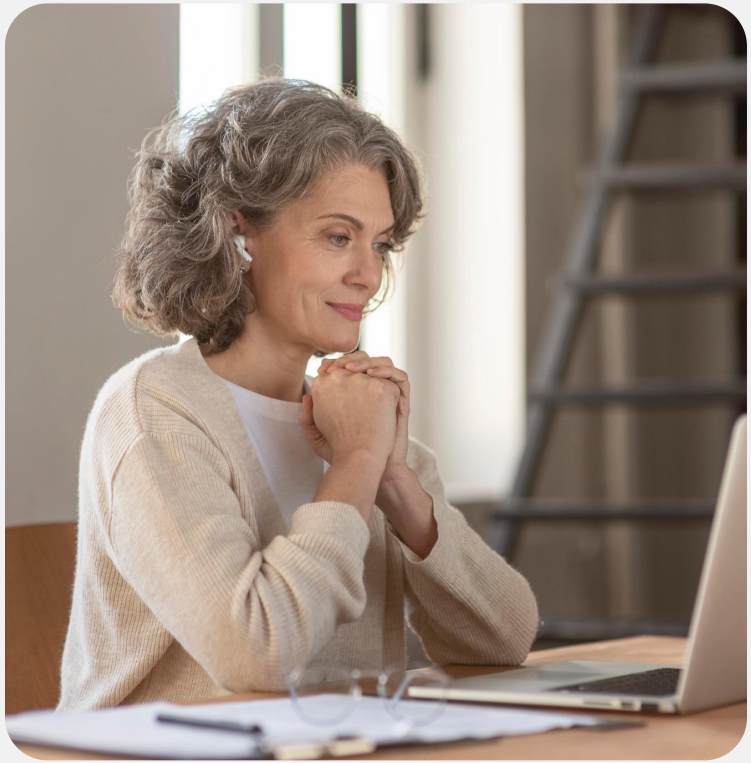


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello, Python!"

print(text[1:6])
```

a. Hello
b. ello,
c. ello
d. Ошибка

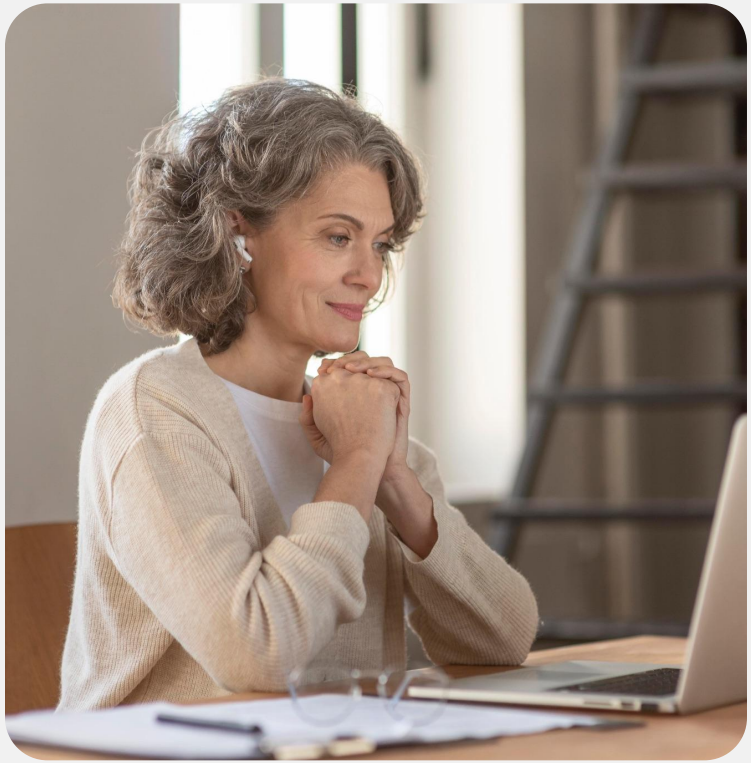


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello"

print(text[10])
```

a. Вывод символа с индексом 10
b. Вывод первого символа
c. Ошибка
d. Вывод последнего символа

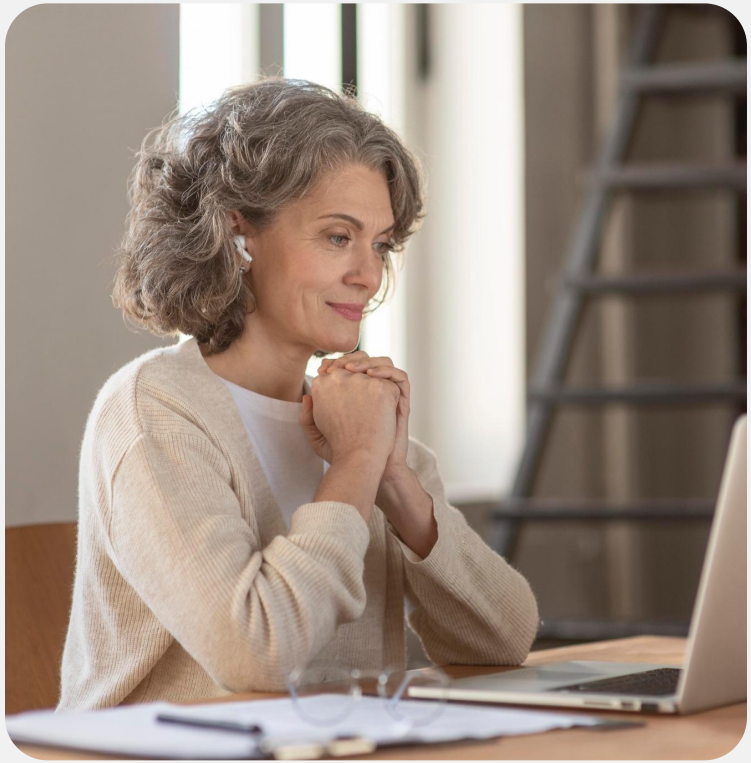


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Hello"

print(text[10])
```

a. Вывод символа с индексом 10
b. Вывод первого символа
c. Ошибка
d. Вывод последнего символа

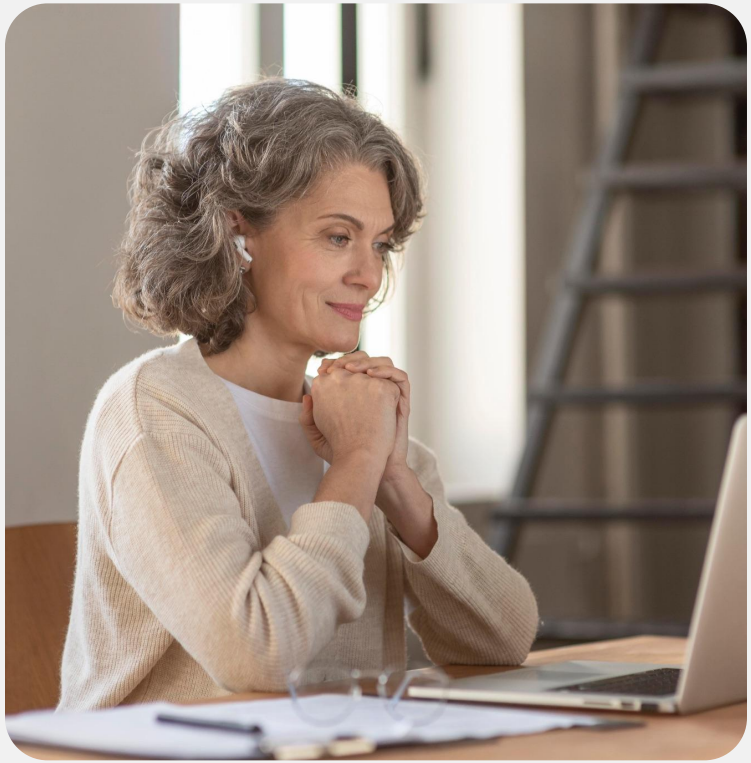


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Python"

print(text[::-1])
```

a. Python
b. nohtyP
c. nohty
d. Ошибка

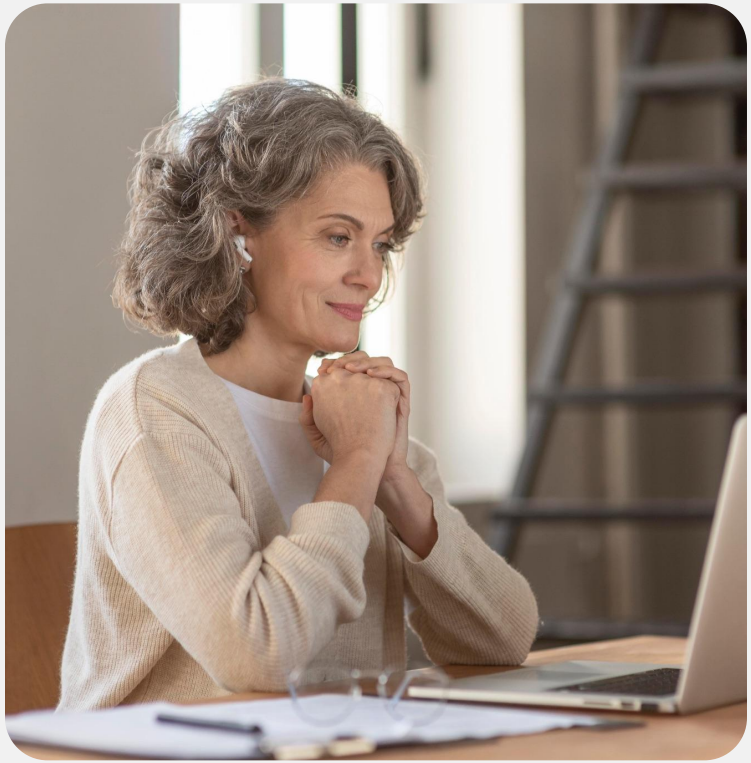


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Python"

print(text[::-1])
```

a. Python
b. nohtyP
c. nohty
d. Ошибка

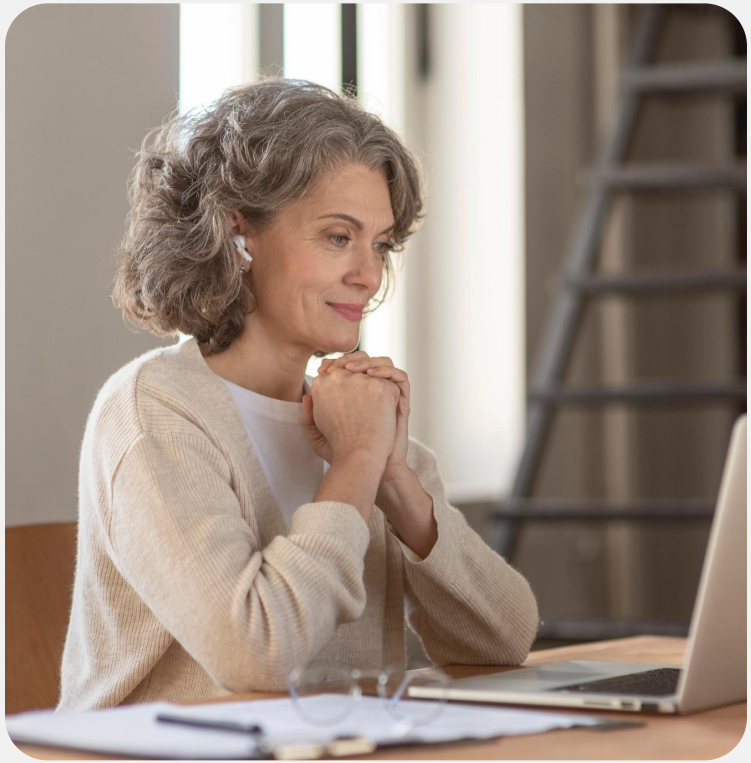


Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Python programming"

print("python" in text)
```

a. True
b. False
c. Ошибка
d. None



Выберите правильный вариант ответа

Какой результат будет выведен?

```
text = "Python programming"

print("python" in text)
```

a. True
b. False
c. Ошибка
d. None

IT CAREER HUB

ПРАКТИЧЕСКАЯ РАБОТА

1. Палиндром

Напишите программу, которая проверяет является ли строка палиндромом. Палиндром - это строка, которая читается одинаково слева направо и справа налево).

Пример вывода:

```
Введите строку: radar
Строка является палиндромом
```

2. Шифр Цезаря

Напишите программу для дешифровки текста, зашифрованного с помощью шифра Цезаря. Программа принимает зашифрованный текст и сдвиг (использованный для зашифровки) от пользователя и выводит расшифрованный текст.

Шифр Цезаря — это один из самых простых и известных методов шифрования. В этом методе каждый символ в исходном тексте заменяется на символ, находящийся на фиксированное число позиций дальше в алфавите. Сдвиг может быть как влево, так и вправо, и для расшифровки необходимо просто сдвинуть символы обратно на тот же шаг.

Пример вывода:

```
Введите зашифрованный текст: khoor
Введите сдвиг: 3
Расшифрованный текст: hello
```

IT CAREER HUB

ДОМАШНЕЕ ЗАДАНИЕ

Домашнее задание

1. **Проверка кодировки**

Напишите программу, которая принимает от пользователя один символ и выводит его код в таблице Unicode и его принадлежность к диапазону ASCII (True/False).

Пример вывода:

```
Введите символ: A

Код Unicode: 65

ASCII символ: True
```

Домашнее задание

2. Подстрока с заполнением

Напишите программу, которая принимает строку и индексы начала и конца. Программа должна вывести подстроку на указанных позициях. Также если конечный индекс выходит за пределы строки, программа заполняет недостающие символы символами _.

Пример вывода:

```
Введите строку: Программирование

Введите начальный индекс: 3

Введите конечный индекс: 20

Подстрока: граммирование_____
```

Заключение