

1. какой текст должен присутствовать в названии тестовых файлов и функций. Требование библиотеки Pytest.

- a. "main" **b. "test_"** c. "_" d. ".py"

2. Название файла, который требуется для корректной работы библиотеки **pytest**

- a. chromedriver.exe b. test_visible.py **c. conftest.py** d. readme.md

3. Какой будет результат теста?

```
def test_example():  
    a = 1  
    assert a in [2, 3, 4]
```

- a. Тест пройден **b. Тест провален** c. Тест пропущен

4. что сделает команда **git commit -m "first commit"**

- a. отправляет изменения в репозиторий.
b. создает коммит с указанным комментарием.
c. инициализирует локальный **git** репозиторий.

5. В приведенном примере представлена реализация популярного алгоритма "сортировка пузырьком".

Какая строка кода содержит оператор условия?

```
bubble_sort.py x  
1  # N ^2  
2  
3  mas = [1, 9, 4, 6, 8, 2, 1]  
4  n = 7  
5  
6  count = 0  
7  for run in range(n-1):  
8      for i in range(n-1-run):  
9          if mas[i] > mas[i+1]:  
10             count += 1  
11             mas[i], mas[i+1] = mas[i+1], mas[i]  
12  
13  print(*mas)  
14  print(count)  
15
```

- a. № 7 b. № 13 c. № 8 d. № 11 **e. № 9**

6. что такое локатор?

- a. метод инициализирующий класс
b. доп строка с типом, ставится после атрибута, указывая на тип которым должен быть этот атрибут.
c. строка, идентифицирующая элемент DOM
d. свойство элемента **DOM**

7. выберите локатор типа **xpath**

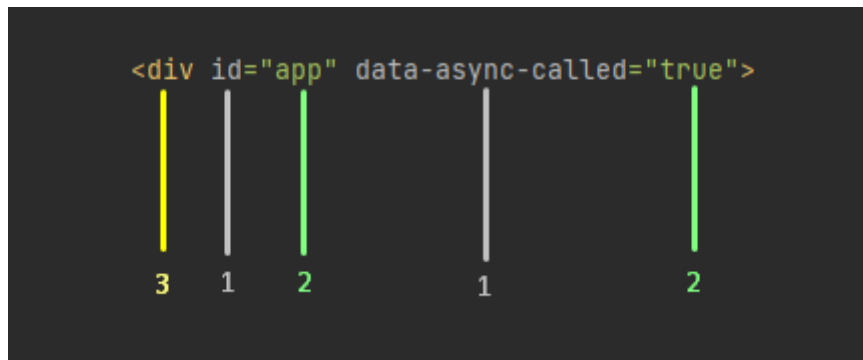
a. #section

☒ b. //div/div/h1

c. div > div.home-body

d. div:nth-child(1)

8. Расставьте соответствие в строении элемента **DOM**



1.	<input checked="" type="radio"/> a
2.	<input checked="" type="radio"/> b
3.	<input checked="" type="radio"/> c

a. Имя элемента

b. Свойство элемента

c. Значение атрибута

d. Имя атрибута

9. Что делает Комбинатор '>', в локаторах типа **CSS_SELECTOR**?

- ☒ a. выбирает только те элементы, которые являются дочерними непосредственно по отношению к указанному элементу.
- b. выбирает элементы, которые находятся внутри указанного элемента (вне зависимости от уровня вложенности).
- c. выбирает элементы, которые находятся на этом же уровне вложенности, после указанного элемента, с тем же родителем.
- d. выбирает элемент, который находится непосредственно после указанного элемента, если у них общий родитель.

10. какие локаторы, из указанных, являются эквивалентными между собой?

- ☒ a. button#size = button[id="size"]
- b. button.login = button[text="login"]
- c. #app = [data-async-called="app"]

11. Чем статический метод, в классе, отличается от обычного?

- a. К статическому методу можно обратиться не создавая экземпляр класса.
- b. В статический метод, в отличии от простого, первым аргументом надо посылать self
- ☒ c. Статические методы выделяются декоратором @classmethod

12. У Вас есть три класса. В какой Вы разместите метод который выполняет следующую функцию:

- Выбирает переданное значение в указанном селекте.

Функция универсальная, срабатывает для любого селекта.

- ☒ a. **WebElement** - в классе собраны методы и атрибуты элементов DOM
- b. **BasePage** - в классе собраны методы и атрибуты любой страницы
- c. **Page(BasePage)** - в классе собраны методы и атрибуты страницы Page

13. Имеется тест кейс:

- зайти на страницу **Page**
- нажать на элемент **elem**
- ввести текст **'hello'** в элемент **input**
- проверить наличие элемента **result**

Выберите блок кода, который реализует этот кейс. Библиотека запуска pytest.

a.

```
def test_main(browser):
    page = Page(browser)
    page.visit()
    page.elem.click()
    page.input.send_keys('hello')
    assert page.result.exists()
```

b.

```
def main(browser):
    page = Page(browser)
    page.visit()
    page.elem.click()
    page.input.send_keys('hello')
    assert page.exists()
```

c.

```
def test_main(browser):
    page = Page(browser)
    page.elem.click()
    page.input.send_keys()
    assert page.result.exists()
```

14. Как называется метод позволяющий выполнять синхронный **JavaScript** на странице?

- a. .elem_javascript() b. .get_dom_attribute() c. .execute_script() d. .set_window_size()

15. Вам необходимо реализовать метод, который очищает весь текст в заданном элементе.
Выберите подходящую реализацию.

```
def clear(self):
    self.find_element().send_keys(Keys.CONTROL + 'a')
    self.find_element().send_keys(Keys.DELETE)
```

a.

```
def clear():
    find_element().send_keys(Keys.CONTROL + 'a')
    find_element().send_keys(Keys.DELETE)
```

b.

```
def clear(self):
    self.find_element().send_keys(Keys.DELETE)
```

c.

```
def clear(self):
    self.find_element().Keys.CONTROL() + 'a'
    self.find_element().Keys.DELETE()
```

d.

16. Выберите метод который устанавливает размеры тестируемого окна.

- a. .window_size(width=W, height=H)
b. webdriver.Chrome(width=W, height=H)
c. .set_window_size(width=W, height=H)
d. driver.window(width=W, height=H)

17. что делает атрибут `driver.current_url` ?

- a. Возвращает результат сравнения текущего урла с переданным
- b.** Возвращает текущий URL
- c. открывает переданный URL в новой вкладке

18. Выберите декоратор библиотеки `pytest`, с помощью которого можно классифицировать тесты.

- a. `@pytest.mark.skip`
- b. `@classmethod`
- c.** `@pytest.mark`

19. Имеется тест кейс:

- послать запрос - метод `query`
- Проверить статус ответа - 201
- Проверить валидность ответа
- Проверить заголовок ответа - 'Content-Type': 'application/json'

Выберите блок кода, который реализует этот кейс. Библиотека запуска `pytest`.

```
def test_example():
    res = api.query()

    assert res.status_code == HTTPStatus.CREATED
    Assert.validate_schema(res.json())
    assert res.headers['Content-Type'] == 'application/json'
```

a.

```
def test_example():
    res = api.query()

    assert res.status_code == HTTPStatus.OK
    Assert.validate_schema(res.json())
    assert res.cookies['Content-Type'] == 'application/json'
```

b.

```
def test_example():
    res = api.query()

    assert res.status_code == HTTPStatus.CREATED
    Assert.validate_schema(res.json())
    assert re.fullmatch(r'\w{1,5}', res.headers['Content-Type'])
```

c.

20. Напишите регулярное выражение под любую дату формата '12 Августа 2023'