Языки скриптинга

Teach Me Skills

Вопросы по предыдущим темам или ДЗ

Teach Me Skills

Mini-quize по новой теме:

1. **Что такое html/css?**
2. **Что такое тэги в веб разработке?**
3. **При помощи чего можно указать стили?**
4. **Что нужно для создания своего сайта с нуля?**
5. **Что такое адаптивный дизайн?**

Teach Me Skills

План занятия

1. Изучим основной синтаксис HTML
2. Так же поработаем с стилями css
3. Поймем основы разработки сайтов и их структуру
4. Изучим сетевые возможности языка python
5. Рассмотрим внешние зависимости для написания полезных скриптов

HTML

HTML — теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными тегами. Тэги могут быть пустыми, не содержащими никакого текста и других данных.

Важно ещё отметить в HTML можно встроить с помощью тега язык программирования JavaScript.
Также включение **CSS** в HTML позволяет задавать внешний вид и макет страницы.

Структура сайта

```
<html>
  <head>
    <title>
      <!-- Your Site Name -->
    </title>

    <style>
      /*Your Site CSS*/
    </style>

    <script>
      //Your Site JS
    </script>
  </head>
  <body>
    <!-- Your Site HTML -->
  </body>
</html>
```

<html></html> Указывает программе просмотра страниц, что это HTML документ.

<head></head> Определяет место, где помещается различная информация не отображаемая в теле документа.

<title></title> Помещает название документа в оглавление программы просмотра страниц

<body></body> Определяет видимую часть документа

Подробнее об HTML

Тэги

Теги форматирования текста
`h1-h6, b, i, tt, cite`

Гиперссылки
`a href`

Формы
`form, select, option, textarea, input`

Таблицы
`Table, tr, td, th`

Графические элементы
`Img, hr`

Форматирование
`p, br`

Единицы измерения: px, e.m., rem
Пиксель –основная еденица измерения!
e.m. – соотносится с размером текста
% - относительно размера родителя!

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Распространенные атаки на HTML

HTML-инъекции

<h1>This is a test</h1>

%3C%68%31%3E%54%68%69%73%20%69%73%20%61%20%74%65%73%74%3C%2F%68%31%3

=

This is test

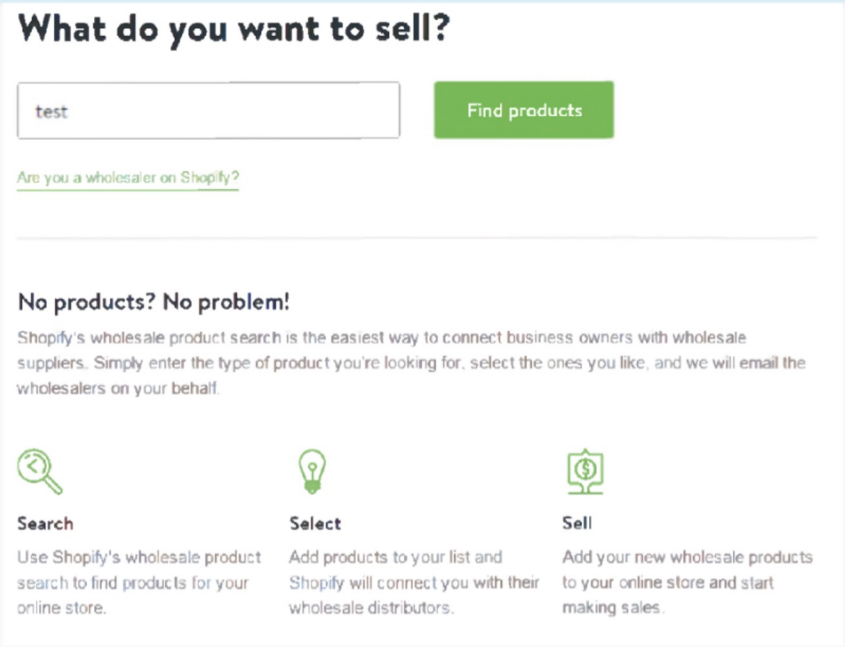
Распространенные атаки на HTML

XSS

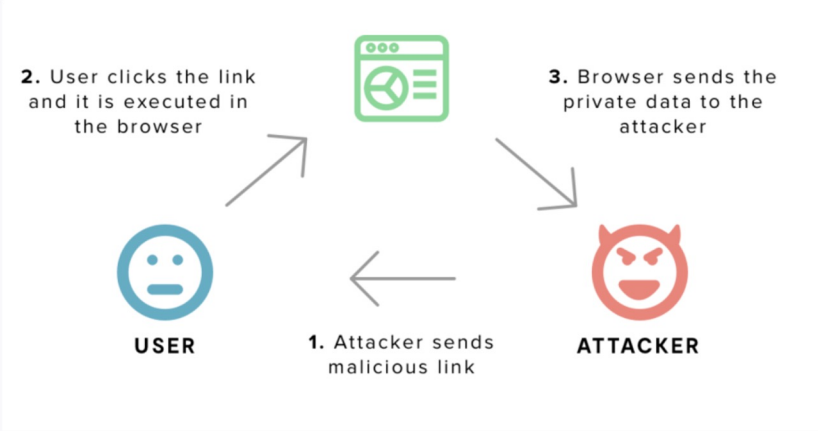
- Отраженные (reflected)
- Хранимые (stored)
- DOM-модели (DOM-based)

Распространенные атаки на HTML xss

Отраженные (reflected)



test<script>alert('XSS');</script>

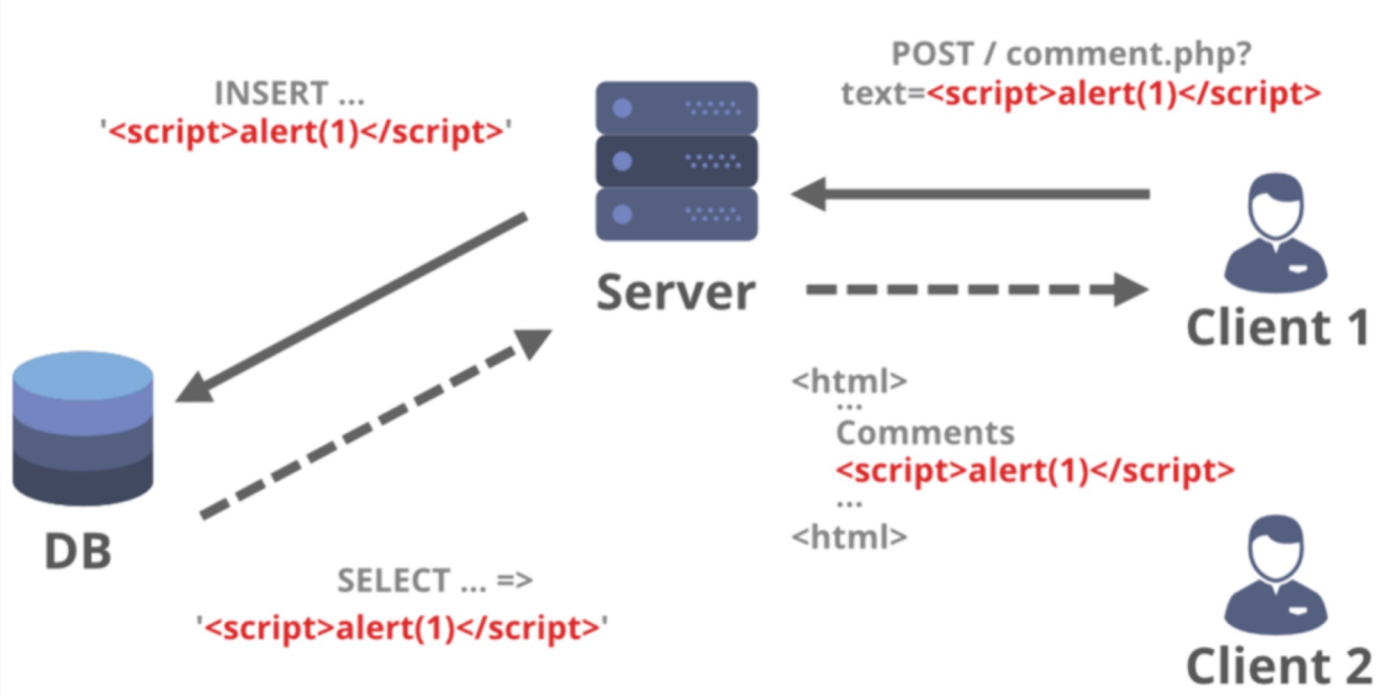


14

vulnsite.com/search/?q=test<script>alert(%27XSS%27);</script>

Распространенные атаки на HTML xss

Хранимые (stored)



Распространенные атаки на HTML XSS

Где может возникнуть XSS

- HTML-атрибут в двойных, одинарных или вообще без кавычек

```
<input type="text" name="some" value="{{user_input}}" />
<input type='text' name='some' value='{{user_input}}' />
<input type=text name=some value={{user_input}} />
```

- data-атрибут

```
<div data-settings='{"next-url":"{{user_input}}"}' >...</div>
```

- URL-адрес

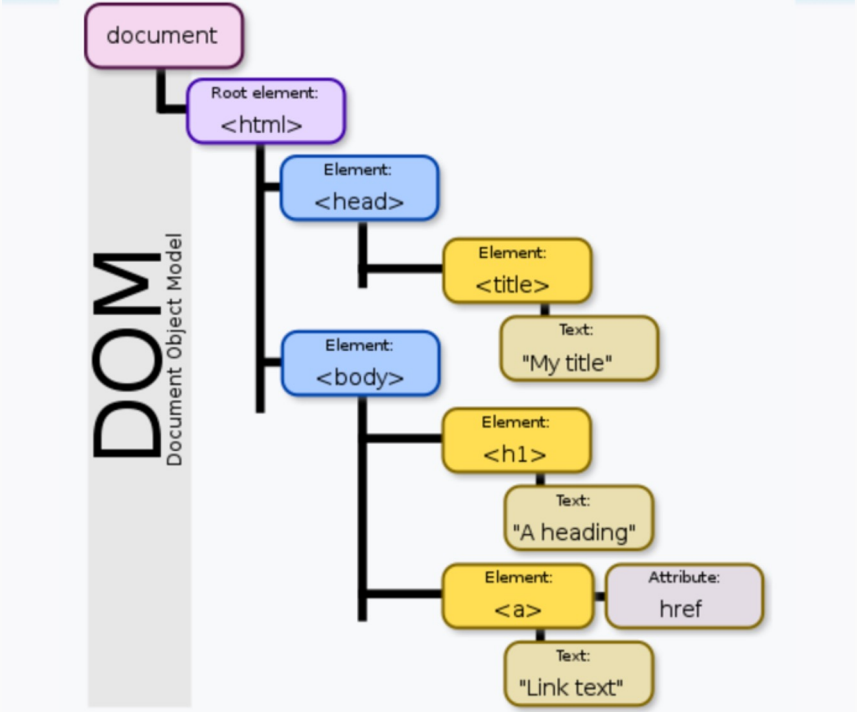
```
<a href="{{user_input}}" rel="noopener">some user link</a>
```

- тег <script>

```
<script>{{user_input}}</script>
```

Распространенные атаки на HTML xss

DOM-модели (DOM-based)



Распространенные атаки на HTML xss

DOM-модели (DOM-based)

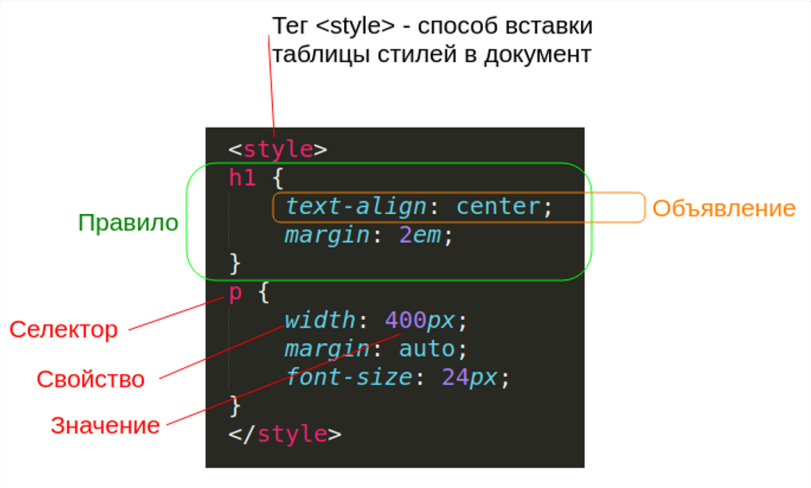
```
<HTML>
<TITLE>Vuln</TITLE>

<SCRIPT>
var pos=document.URL.indexOf("name=")+5;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>

Welcome!
</HTML>
```

http://www.vuln.com/index.html?name=<script>alert(document.cookie)</script>

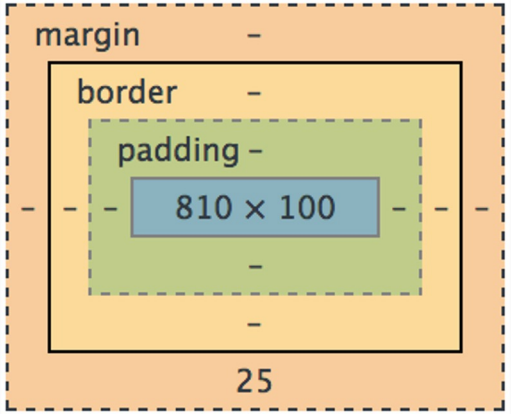
CSS

CSS - это язык таблицы стилей. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML.

Синтаксис стилей CSS:



Боксовая модель CSS:



CSS

color:	изменение цвет текста
background-color:	изменение цвет фона
background-image:	изображение, в качестве фона фона
font-size:	размер шрифта
font-family:	имя шрифта
font-style:	стиль шрифта
font-weight:	толщина шрифта
text-align:	выравнивание текста
text-indent:	отступ первой строки
text-shadow:	тень текста
border-color:	цвет границ
border-width:	толщина границ
border-style:	стиль границ
margin:	отступ между краем фона одного блока и краем фона другого блока
padding:	отступ между краем фона элемента и его содержимым
width:	ширина
height:	высота

Python

Что такое Python

Python – это высокоуровневый язык программирования, который был разработан в конце 1980-х годов. Его разработчик, Гвидо ван Россум, вложил в основу языка простоту и читабельность кода, что позволяет использовать Python для быстрой и эффективной разработки. Много популярных веб-сайтов, компьютерных игр и программ, написанных на Python, вы используете ежедневно: Dropbox, Uber, Sims, Google, GIMP и другие.

Язык отличается понятным синтаксисом, поэтому Python подходит для начинающих программистов. Он широко используется во многих областях: веб-разработка, научные исследования, анализ данных, искусственный интеллект, машинное обучение, разработка игр.

У Python большая библиотека сторонних модулей и инструментов, что делает его мощным инструментом. Наличие активного сообщества разработчиков позволяет постоянно поддерживать и обновлять язык, предоставлять достаточный объем обучающих материалов, документацию и форумы для программистов с любым уровнем знаний.

Преимущества Python

1. Простой и читаемый код. Python предлагает понятный синтаксис, что делает его привлекательным для опытных разработчиков и доступным для новичков.
2. Большое число полезных библиотек и модулей для Python позволяет быстро и легко решать различные задачи, такие как обработка данных, машинное обучение, работа с базами данных.
3. Язык подходит для большинства операционных систем. Код, написанный при помощи Python, может быть запущен на популярных ОС: Windows, macOS, Linux.
4. Python позволяет легко интегрировать код на других языках, таких как C ++ и Java. Это позволяет использовать уже существующий код и библиотеки на этих языках, чтобы расширять функциональность Python.
5. Активное сообщество разработчиков помогающее и поддерживающее новичков. Это значит, что всегда можно получить ответы на возникающие вопросы или найти готовый код для решения своих задач.

Основные свойства и возможности Python

1. Python интерпретируемый язык программирования – код на нем выполняется построчно, в режиме реального времени. Это свойство позволяет быстро исправлять и проверять код без необходимости компиляции.
2. Python является языком с динамической типизацией, то есть тип переменной определяется автоматически, во время выполнения кода. Это упрощает процесс программирования и делает его гибким при работе с данными различного типа.
3. Python поддерживает объектно-ориентированное программирование (ООП), что позволяет разрабатывать код в виде объектов, которые взаимодействуют друг с другом. Это делает код более модульным и повторно используемым.
4. Python поддерживает также императивное, функциональное и аспектно-ориентированное программирование. Таким образом, разработчики имеют возможность выбирать нужный подход для решения конкретной задачи.

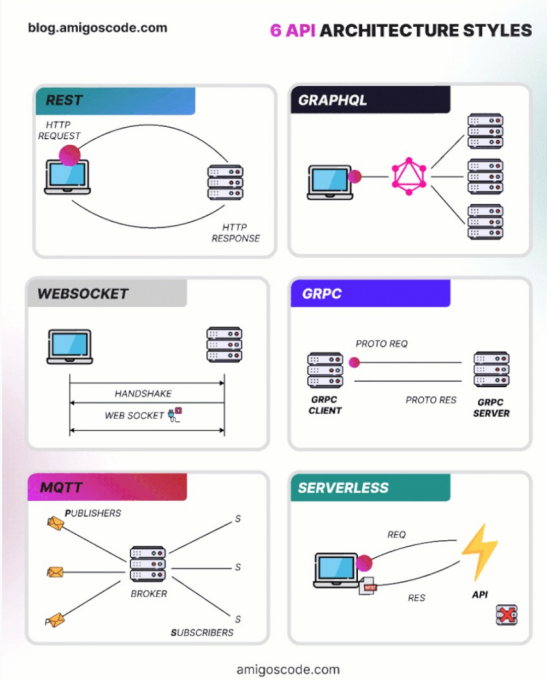
Python

Основы python

Запросы

Teach Me Skills

Типы архитектур запросов



[REST](#) — архитектурный стиль взаимодействия компонентов распределённого приложения в сети, набор правил того, как программисту организовать написание кода серверного приложения, чтобы все системы легко обменивались данными и приложение можно было масштабировать

[WebSocket](#) - протокол для общения между клиентом и сервером, предоставляющий двухсторонне общение сверх протокола TCP.

[MQTT](#) — упрощённый сетевой протокол, работающий поверх TCP/IP, ориентированный на обмен сообщениями между устройствами по принципу издатель-подписчик.

[Serverless](#) — стратегия организации платформенных облачных услуг, при которой облако автоматически и динамически управляет выделением вычислительных ресурсов в зависимости от пользовательской нагрузки.

[gRPC](#) google **Rem Proc Call`** — это система удалённого вызова процедур (RPC) с открытым исходным кодом, первоначально разработанная в Google в 2015 году.

[GRAPHQL](#) - это язык запросов и серверная среда для API с открытым исходным кодом.

JSON

Teach Me Skills

JSON — текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Но при этом формат независим от JS и может использоваться в любом языке программирования.

```
{
    "name": "John",
    "age": 30,
    "city": "New York",
    "address": {
        "street": "123 Main St",
        "city": "Anytown",
        "postalCode": "12345"
    },
    "data": ["Hello", "from", "browser"]
}
```

Структура данных: JSON предоставляет структурированный формат данных, который поддерживает следующие базовые типы:
Объект: Набор пар ключ-значение, заключенных в фигурные скобки {}. Ключи должны быть строками, а значения могут быть строками, числами, логическими значениями, объектами, массивами или null.

Массив (Array): Упорядоченный список значений, заключенных в квадратные скобки **[]**.
Так же значения которые могут принимать данные: int, string, bool, null.

JSON часто используется для передачи данных между веб-сервером и клиентом. Например, API может возвращать данные в формате JSON, который затем обрабатывается на стороне клиента.

GET & POST

GET — метод для чтения данных с сайта, для доступа к указанной странице. Говорит серверу, что клиент хочет прочитать указанный документ. Фильтры, которые выбирает пользователь, передаются через метод GET.

1.Параметры: Параметры запроса, передаются в URL после вопросительного знака (**?**).
Например: **http://example.com/path?name=value**.
2.Ограничение длины: Длина URL ограничена, около 2000-2048 символов.
3.Кэширование: GET-запросы могут быть закэшированы.
4.Безопасность: GET- менее безопасные, параметры передаются в URL и видны в строке браузера и логах.
5.Использование: GET-запросы обычно используются для получения данных с сервера.

POST — метод для отправки данных на сайт. С помощью метода POST передаются формы.

Параметры в теле запроса: Параметры запроса передаются в теле HTTP-запроса, и не отображаются в URL.
Длина запроса: Длина POST-запроса не ограничена длиной URL, и они могут использоваться для передачи больших объемов данных.
Некэшируемость: POST-запросы не кэшируются.
Безопасность: POST-запросы считаются более безопасными, поскольку параметры не отображаются в URL и не видны в логах.
Использование: POST-запросы обычно используются для отправки данных на сервер, например, при отправке форм на веб-страницах, и они не подходят для закладок браузера

Запросы библиотекой requests

Teach Me Skills

```
pip install requests
```

```python
import requests

url = 'https://www.example.com/search'
headers = {'User-Agent': 'my-app'}
params = {'q': 'python', 'page': 1}

response = requests.post(url, headers=headers, params=params)

print(params)
```

Значения кодов состояния:
- 1XX — информация
- 2XX — успешно
- 3XX — перенаправление
- 4XX — ошибка клиента (ошибка на вашей стороне)
- 5XX — ошибка сервера (ошибка на их стороне)

Teach
Me
Skills

Спасибо за внимание!