

HTML и принципы автотестирования веб-приложений

Sumy Educational Center
Software Quality Assurance (QA)

HyperText (гипертекст)



Ted Nelson
Xanadu Project, 1963

Hypermedia → Hypertext → Hyperlinks

Сферы применения концепции гипертекста:

- литература;
- документоведение;
- программное обеспечение;
- Internet (1969)
- WWW (1992)

Типы гиперссылок:

- inline (часть контента донора встроена в контент реципиента);
- anchor (контент реципиента только указывает на нужную часть контента донора)

Markup Language (язык разметки)

TEX

HTML



```
<?xml version="1.0"?>
<quiz>
  <question>
    Who was the forty-second
    president of the U.S.A.?
  </question>
  <answer>
    William Jefferson Clinton
  </answer>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML



Adobe

WIKIPEDIA

PostScript 3

for Office Printing

Типы разметок:

- презентационная разметка (текстовые редакторы типа WYSIWYG и WYSIWYM);
- процедурная разметка (LaTeX, PostScript, др.);
- семантическая разметка (HTML, XML, др.);

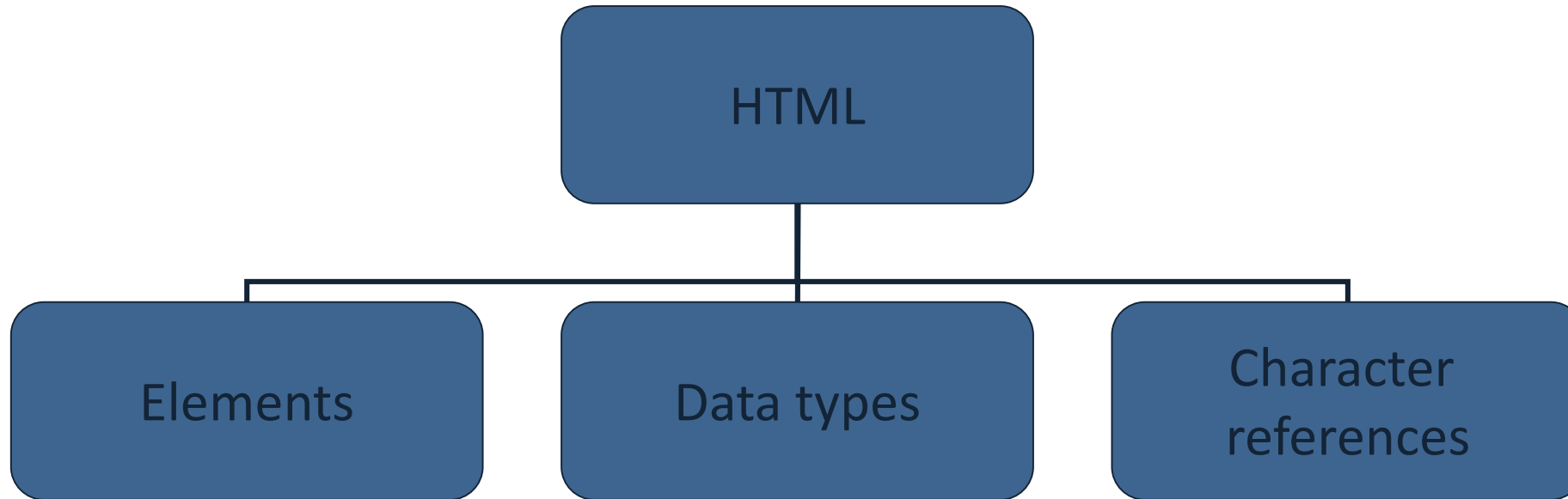
HyperText Markup Language



Tim Berners-Lee
CERN, 1990

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Структура языка HTML



- **Start tag** <tag>
 - **Attributes**
- **End tag**
- **Content**

- **for attributes:**
 - **IDs, names, URIs, colors...**
- **for content:**
 - **script or stylesheet**

- **escape**-последовательности:
 - **< >**
 - **&**
 - **"**
 - **др.**

Элементы и атрибуты

Общий синтаксис описания элементов:

- содержит контент;
- содержит закрывающий тег

```
<tag attribute1="value1" attribute2="value2">  
  content  
</tag>
```

Пример:

```
<a href="http://www.netcracker.com/"> NetCracker </a>
```

Синтаксис описания пустых элементов

- не содержит контент;
- не содержит закрывающий тег

```
<tag attribute1="value1" attribute2="value2">
```

Пример:

```
<IMG src="image.gif" alt="descriptive text" width="50" height="50" border="0">
```

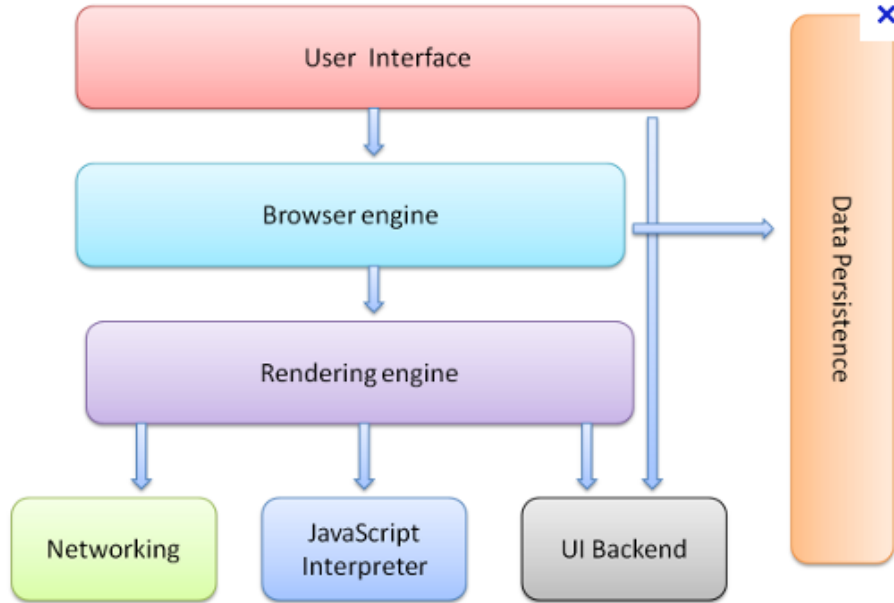
Элементы и атрибуты

Элементы HTML:

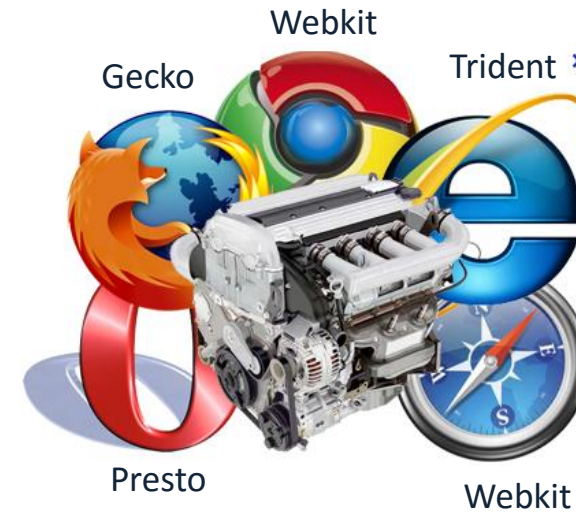
- 1) Structure elements (html, head, body)
- 2) Head elements (title, meta, style, ...)
- 3) Body elements:
 - 1) Block elements (h1,..., h6, p, li, div, ...)
 - 2) Inline elements (a, b, i, u, br, q, cite, ...)
 - 3) Images and objects (img, object)
 - 4) Forms (form, button, input, ...)
 - 5) Tables (table, tr, td, col, ...)
 - 6) Frames (frame, iframe, ...)

Принцип работы WEB браузеров

Основные компоненты браузеров



Движки браузеров

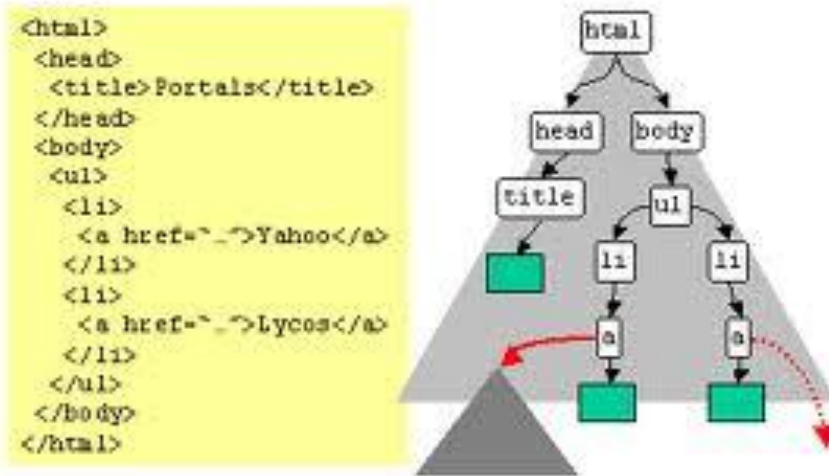


Основные этапы рендеринга



Принцип автотестирования WEB приложений

DOM W3C®



Одной из основных операций при автоматизированном тестировании является **задача управления html-документами в браузере** – загрузка, нажатия на ссылки, установка значений, чтение состояния элементов и т.д., т.е. повторение тех действий, которые выполняет человек при работе с web-приложением.

Управление WEB контентом происходит через **объектную модель документа** (DOM - document object model).

Два backend'а: Selenium и HTMLUnit



Selenium – это комплексный набор разнообразных инструментов для тестирования web-приложений в браузерах, которые поддерживают Javascript.

Компоненты Selenium:

- Selenium IDE
- Selenium Remote Control (Selenium RC) - backend
- Selenium WebDriver

«+»: работа с AJAX, хорошо эмулирует работу пользователя.

«-»: медленный, работает только с OS Windows



HtmlUnit

GUI-Less browser for Java programs

HTMLUnit – это java-библиотека, которая эмулирует браузер; она моделирует html-документ и предоставляет API для управления этим документом (заполнения элементов, нажатие ссылок и т.д.).

По сути HTMLUnit – это браузер без графической оболочки, благодаря чему он работает в несколько раз быстрее Selenium.

«+»: быстрота, кроссплатформенность

«-»: плохо работает с AJAX, не всегда достоверно эмулирует действия пользователя

Идентификация элементов

Локатор (locator) – строка, уникально идентифицирующая UI элемент

Типы локаторов по приоритетам использования (критерий перехода – невозможность уникально идентифицировать элемент):

- 1) link-локатор;
- 2) id-локатор;
- 3) name-локатор;
- 4) dom-локатор;
- 5) css-локатор;
- 6) x-path-локатор

Идентификация элементов. Link-локатор

Link-локатор используется только для идентификации гиперссылок

HTML-код:

```
<a href='http://some_url'>
```

```
  Link Text
```

```
</a>
```

Локатор в Selenium:

Link = Link Text

HTML-код:

```
<a href='http://some_url'>
```

```
  Link Text Feb 18, 2012
```

```
</a>
```

Локатор в Selenium:

Link = Link Text *

Идентификация элементов. ID-локатор

ID-локатор используется для идентификации любых элементов по их атрибуту ID.

Преимущество: быстрота.

Недостаток: атрибут ID часто не указывается

HTML-код:

```
<input  
  type=text  
  id='some_input_id'  
  name='some_input_name'  
  value=''  
>
```

Локатор в Selenium:

```
id = some_input_id
```

Идентификация элементов. Name-локатор

Name-локатор используется для идентификации любых элементов по их атрибуту name.

Преимущество: удобен для полей ввода формы (кнопки, текстовые поля, выпадающие списки).

Недостаток: менее уникальный, чем атрибут ID

HTML-код:

```
<input  
  type=text  
  id='some_input_id'  
  name='some_input_name'  
  value=""  
>
```

Локатор в Selenium:

```
name = some_input_name
```

Идентификация элементов. DOM-локатор

DOM-локатор используется для идентификации любых элементов по их месту в DOM-структуре документа.

HTML-код:

```
<form name=first_form>
  <input
    class=sample_class type=text name=sample_name
  />
</form>
<form name=second_form>
  <input
    class=sample_class type=text name=sample_name
  />
</form>
```

Локатор в Selenium:

dom=document.second_form.sample_name

```
|-> Document
  |-> Element (<html>)
    |-> Element (<body>)
      |-> Element (<div>)
        |-> text node
        |-> Anchor
          |-> text node
        |-> Form
          |-> Text-box
          |-> Text Area
          |-> Radio Button
          |-> Check Box
          |-> Select
          |-> Button
```

Идентификация элементов. CSS-локатор

CSS-локатор используется для идентификации любых элементов по их месту в каскадной таблице стилей.

HTML-код:

```
<form name=first_form>
  <input
    class=sample_class type=text
    name=j0122:sample_name
  />
</form>
<form name=second_form>
  <input
    class=sample_class type=text
    name=j0123:sample_name
  />
</form>
```

```
css=form[name=second_form]
css=input[name~=sample_name]
css=form[name=second_form] >
  input[name~=sample_name]
```


Идентификация элементов. XPath-локатор

XPath-локатор используется для идентификации любых элементов с помощью синтаксиса языка XPath (XML Path Language).

HTML-код:

```
<form name=first_form>
  <input
    class=sample_class type=text
    name=j0122:sample_name
  />
</form>
<form name=second_form>
  <input
    class=sample_class type=text
    name=j0123:sample_name
  />
</form>
```

Локатор в Selenium:

```
xpath=//form[@name='second_form']
xpath=//input[contains(@name,'sample_name')]
xpath=//form[@name='second_form']/
      input[contains(@name,'sample_name')]
```

Q&A

Thank You

