

# Web-технологии 3

Определение типа документа  
(DTD, Document Type Definition)

# Введение

- Процесс формального определения языка — моделирование документа
- модель документа устанавливает согласование документа с языком
- модель — документ особого рода, описывает словарь и разметку
- согласующиеся документы — действительные (valid), остальные — недействительные (invalid)

# Зачем нужна модель?

- Документы создаются людьми для машин
- наличие обязательных полей (цена, вес)
- получение документов со сторонних источников
- определение надежной структуры языка

# Проблемы

- Дополнительные затраты на сопровождение
- замедление работы
- ограничение использования элементов
- возможные проблемы с пространствами имен

Если не использовать модель

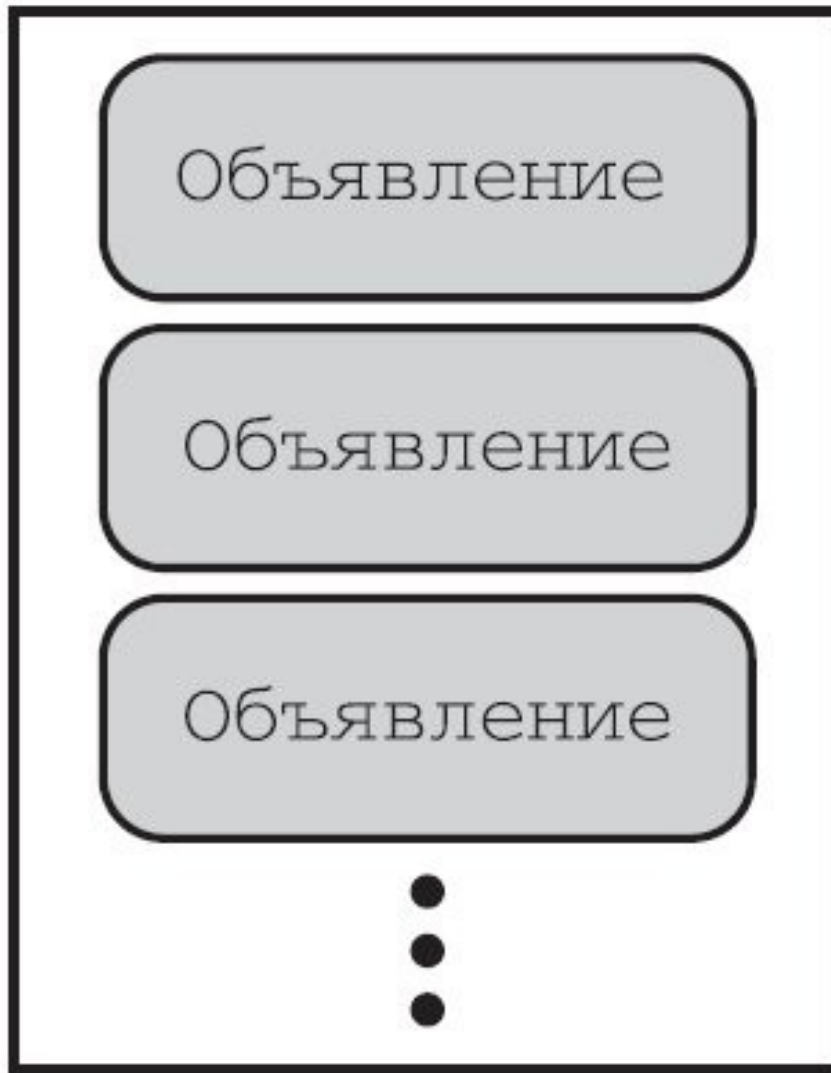
- неограниченный словарь
- отсутствие правил
- отсутствие ограничений на атрибуты

# DTD

- Реализация модели в виде определения типа документа (DTD, Document Type Defenition)
- DTD основан на SGML
- Использование:
  - объявление множества символов (словарь)
  - определение модели содержимого каждого элемента
  - определение набора разрешенных атрибутов
  - дополнительные механизмы управления моделью

# Синтаксис DTD

- DTD — набор правил (объявлений)



- Порядок имеет значение если:
  - указаны двойные объявления (берется первое)
  - если указаны параметрические сущности, то они д.б. объявлены выше

# Объявление элемента

- новый элемент в пространстве языка
- `<!ELEMENT name content-model>`
  - `name` — имя элемента
  - `content-model` — модель содержимого
- Модели:
  - Пустой элемент
    - `<!ELEMENT graphic EMPTY>`
  - без ограничений на содержимое
    - `<!ELEMENT contain-anything ALL>`



- СИМВОЛЬНЫЕ ДАННЫЕ
  - <!ELEMENT emphasis (#PCDATA)>
- ТОЛЬКО ЭЛЕМЕНТЫ
  - <!ELEMENT article (title, (para | sect1)+)>
- СМЕШАННОЕ СОДЕРЖИМОЕ
  - <!ELEMENT para (#PCDATA | emphasis | xref)\*>

# Частота присутствия

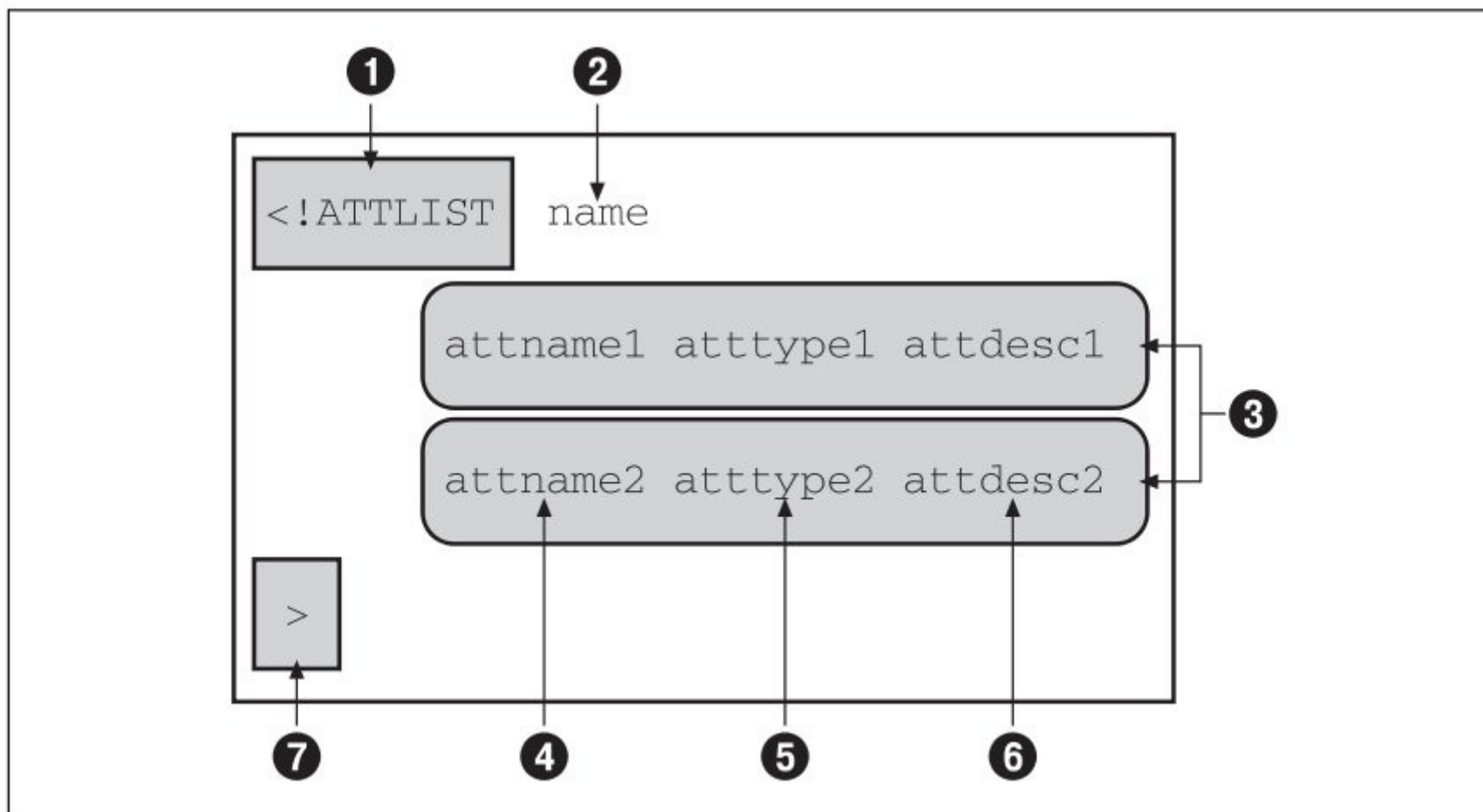
- "," - необходимая последовательность (AND)  
A,B — после A идет B а не наоборот
- "|" - альтернативы (OR)
- "(...)" - объединение содержимого
- "?" - необязательность предыдущего элемента
- "+" - одно или больше присутствий
- "\*" - любое количество присутствий

# Пример

<!ELEMENT article (title, subtitle?, author\*, (para | table | list)+, bibliography?) >

- Допустимые варианты article:
  - title, subtitle, author, para, para, para
  - title, author, author, para, list, para, table, para, bibliography
  - title, list, list, list, bibliography
  - title, subtitle, table, para

# Объявление атрибутов



1 — начальная строка объявления  
2 — имя элемента  
3 — объявления атрибутов  
7 — завершающий разделитель

4 — имя атрибута  
5 — тип или значения  
6 — режим атрибута

# Типы атрибутов

- CDATA — символьные данные
  - `<!ATTLIST circle-radius CDATA "12 inches">`
- NMTOKEN — метка имени
- NMTOKENS — список меток имен разделенных пробелами
- ID — уникальный идентификатор
- IDREF — ссылка на идентификатор
- IDREFS — список ссылок на идентификаторы

- ENTITY — имя сущности
  - <!ATTLIST bulletlist icon ENTITY #IMPLIED>  
<!ENTITY bluedot SYSTEM "icons/bluedot.png">
  - Использование:
    - <bulletlist icon="bluedot">
- ENTITIES — список имен сущностей
- список перечисляемых значений  
разделенных "|"
  - <!ATTLIST part-instock ( true | false ) #IMPLIED>
- NOTATION — список нотаций

# Режимы атрибутов

- значение по умолчанию
  - `<!ATTLIST message-importance (high | medium | low) "medium">`
- необязательный (`#IMPLIED`)
- обязательный (`#REQUIRED`)
- зафиксированное значение (`#FIXED`)

# Пример

```
<?xml version="1.0"?>
<!DOCTYPE checkbook SYSTEM "checkbook.dtd">
<checkbook>

  <deposit type="direct deposit">
    <payor>Bob's Bolts</payor>
    <amount>987.32</amount>
    <date>21 6 00</date>
    <description category="income">Paycheck</description>
  </deposit>

  <payment type="check" number="980">
    <payee>Kimora's Sports Equipment</payee>
    <amount>132.77</amount>
    <date>23 6 00</date>
    <description category="entertainment">Kendo
equipment</description>
  </payment>
```



```
<payment type="debit">
  <payee>Lone Star Cafe</payee>
  <amount>36.86</amount>
  <date>26 6 00</date>
  <description category="food">Lunch with Greg</description>
</payment>
```

```
<payment type="check" number="981">
  <payee>Wild Oats Market</payee>
  <amount>47.28</amount>
  <date>29 6 00</date>
  <description category="food">Groceries</description>
</payment>
```

```
<payment type="debit">
  <payee>Barnes and Noble</payee>
  <amount>58.79</amount>
  <date>30 6 00</date>
  <description category="work">O'Reilly Books</description>
</payment>
```

```
</checkbook>
```

- Корневой элемент <checkbook>
- Содержит записи <payment> (платеж) и <deposit>(вклад):

<!ELEMENT checkbook (deposit | payment)\*>

- Дочерние элементы <deposit>: <payor>, <amount>, <date> и <description> (плательщик, сумма, дата и описание).

<!ELEMENT deposit (payor, amount, date, description?)>

ИЛИ

<!ELEMENT deposit (  
 ((amount, ((date, payor) | (payor, date))) |  
 (date, ((amount, payor) | (payor, amount))) |  
 (payor, ((amount, date) | (date, amount))), description)>

- У элемента <deposit> атрибут type:

<!ATTLIST deposit type #CDATA #IMPLIED>

- Элемент <payment> аналогичен <deposit>:

<!ELEMENT payment (payee?, amount, date, description?)>  
<!ATTLIST payment type (atm | check | debit) #REQUIRED>

- Остальные элементы:

<!ELEMENT amount (#PCDATA)\*>

<!ELEMENT date (#PCDATA)\*>

<!ELEMENT description (#PCDATA)\*>

<!ELEMENT payee (#PCDATA)\*>

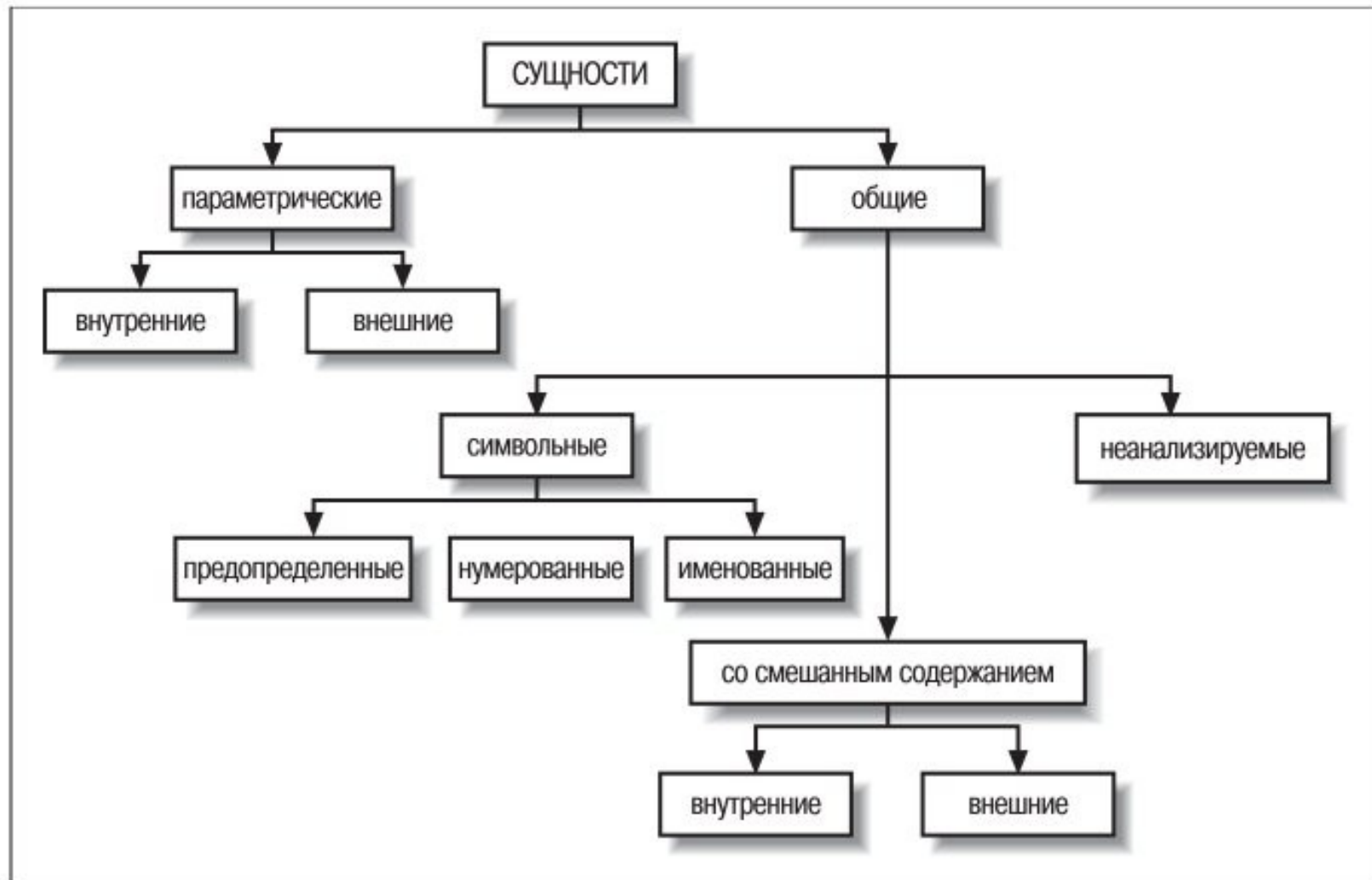
<!ELEMENT payor (#PCDATA)\*>

<!ATTLIST description

category (cash | entertainment | food | income | work) 'food'>

# Сущности (entity)

- Заместитель (placeholder) содержания
- Однажды объявить и многократно использовать



- Сущность = имя + значение
- Определяются в объявлениях (declarations)
- В документе ссылки на сущности (entity references)
- Ссылка на сущность он заменяется значением
- Анализ возобновляется с точки перед местом замены  
=> новый текст тоже анализируется
- Синтаксис ссылок:

`&name;` <!-- общие сущности -->

`%name;` <!-- параметрические сущности -->

- Все сущности (кроме предопределенных) должны быть объявлены раньше, чем они использованы в документе.
- Пример объявления общей сущности:

```
<!ENTITY phone "<number>617 555 1299</number>">
```

- Использование:

```
<body> To reserve a place call at &phone;.</body>
```

- Символьные сущности (character entities) - содержащие единственный символ

- Нумерованная символьная сущность:

&#241; ~ ñ

- Предопределенные символьные сущности:

&amp; ~ &

&apos; ~ '

&gt; ~ >

&lt; ~ <

&quot; ~ "

- Именованные символьные сущности:

&THORN; ~ Þ

# Сущности со смешанным содержанием

- Внутренние сущности (internal entities): текст определяется в объявлении сущности;

```
<!ENTITY bobco "Bob's Bolt Bazaar, Inc.">
```

- Внешние сущности (external entities): текст находится в другом файле

```
<!ENTITY part1 SYSTEM "p1.xml">
```

```
<!ENTITY catalog SYSTEM
```

```
"http://www.bobsbolts.com/catalog.xml">
```

```
<!ENTITY faraway PUBLIC " //BOB//FILE Catalog//EN"
```

```
"http://www.bobsbolts.com/catalog.xml">
```

- Не анализируемые сущности (unparsed entity)

```
<!ENTITY mypic SYSTEM "photos/erik.gif" NDATA GIF>
```

# Параметрические сущности (parameter entity)

- содержит текст из DTD
- может использоваться во внутреннем или внешнем подмножестве
- не может содержать текст XML,
- ссылка на параметрическую сущность не может находиться внутри документа XML.

```
<!ENTITY % content "para | note | warning">
```

```
<!ELEMENT chapter (title, epigraph, %content;+)>
```

- Результат:

```
<!ELEMENT chapter (title, epigraph, para | note | warning+)>
```

- Внешняя параметрическая сущность

```
<!ENTITY % inline-elements SYSTEM "inlines.mod">
```

```
%inline-elements;
```