

به صورت کلی راجع به ابزار های مورد استفاده :

از مرورگر کروم ، نوت پد ++ و Liquid Studio 2020 به صورت لوکال و از برخی سایت های اینترنتی استفاده شده است.

1 : تشکیل employees.dtd DTD .

این فایل ایجاد شد و کد های آن نوشته شد ، با توجه به نکاتی نظیر اسم تگ ها و رابطه بین تگ ها ، تایپ ها و یا عباراتی که تگ ها آنها را در درون خود حاوی هستند.

```
employees.dtd
1  <?xml encoding="UTF-8"?>
2
3  <!ELEMENT employees (employee)+>
4  <!-- ATTLIST employees -->
5  xmlns CDATA #FIXED ''
6
7  <!ELEMENT employee (name,age?,phone+,hiredate,devices)>
8  <!-- ATTLIST employee -->
9  xmlns CDATA #FIXED ''
10 code NMTOKEN #REQUIRED
11
12 <!-- ELEMENT name (lastname,firstname) -->
13 <!-- ATTLIST name -->
14 xmlns CDATA #FIXED ''
15
16 <!-- ELEMENT age (#PCDATA) -->
17 <!-- ATTLIST age -->
18 xmlns CDATA #FIXED ''
19
20 <!-- ELEMENT phone (#PCDATA) -->
21 <!-- ATTLIST phone -->
22 xmlns CDATA #FIXED ''
23
24 <!-- ELEMENT hiredate (#PCDATA) -->
25 <!-- ATTLIST hiredate -->
26 xmlns CDATA #FIXED ''
27
```

در تصویر بالا لاین هایی از کد را می بینید که به توضیح برخی قسمت ها می پردازیم .

لاین اول الزامیست و تایپ UTF-8 برای پردازش این کد ها باید حتما تنظیم شود.

لاین سوم بیان گر این مسئله است که تگ اصلی ما employees میباشد و باقی رکورد ها به صورت تکرار شونده تحت عنوان employee وجود دارند .

لاین هفتم بیانگر تگهای درون employee است که هر کدام اطلاعات مربوط به تگ خود را نگه داری میکنند.

لاین دوازدهم بیانگر زیر مجموعه های تگ name است که خود name زیر مجموعه employee میباشد.

لاین دهم بیان گر این مسئله است که code به شکل زیر باید درون employee مقدار دهی شود.

```
<?xml version="1.0" encoding="UTF-8"?>
<employees>
  <employee code="e1">
```

NMTOKEN (نشانه نام) هر ترکیبی از کاراکترهای نام است. ... در حالی که نام ها محدودیت هایی در کاراکتر اولیه دارند (اولین کاراکتر یک نام نمی تواند شامل اعداد، نشانه ها، نقطه پایان و خط فاصله باشد)، NMTOKEN این محدودیت ها را ندارد.

CDATA به این معنی است که عنصر حاوی داده های کاراکتری است که قرار نیست توسط یک تجزیه کننده تجزیه شود. #PCDATA به این معنی است که عنصر حاوی داده هایی است که قرار است توسط یک تجزیه کننده تجزیه شود. کلمه کلیدی ANY یک عنصر را با هر محتوایی اعلام می کند. اگر یک بخش #PCDATA حاوی عناصر باشد، این عناصر نیز باید اعلام شوند.

بحث مربوط به #FIXED :

مقدار مشخصه در DTD ثابت میشود و نمی توان آن را تغییر داد یا در سند XML لغو کرد. ... یک مقدار پیش فرض یا ثابت. تجزیه کننده این مقدار را در سند XML درج می کند که مشخصه از دست رفته یا در سند XML استفاده نشده باشد.

Type	Description
CDATA	The value is character data
(en1 en2 ..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

Value	Explanation
<i>value</i>	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED <i>value</i>	The attribute value is fixed

وقتی گزینه ای برای مقدار پیش فرض نداریم، اما همچنان می خواهیم خصیصه را مجبور کنیم که وجود داشته باشد، از کلمه کلیدی REQUIRED# استفاده میکنیم.

2 : تشکیل employees_dtd.xml که اعلان اتصال به DTD خارجی دارد.

```
<!DOCTYPE note SYSTEM "employees.dtd">
```

قطعه کد بالا باید به کد xml اضافه گردد تا بتواند فایل dtd ایجاد شده را فراخوانی کند و به آن متصل شود.

XML Validation

☐XPath ☒Schema ☐DTD

Charset:

Choose XML file to validate: employees_dtd.xml

XML Schema: employees.dtd

Validation Result

0 Error(s)

تصویر بالا حاوی تست dtd و xml داده شده است و همانطور که مشهود است هیچ گونه اروری پیدا نشده است. این اعتبار سنجی پیش از اعمال محدودیت بر روی price و اعمال بازه 20 تا 3000 است.

<http://xmlvalidator.new-studio.org/>

3 : تشکیل employees_dtd_embed.xml که تعریف DTD را درون خود دارد.

برای انجام این کار کفایست کدهای dtd را طبق فرمت زیر درون فایل xml جای گذاری کنیم.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE employees
[ "put the DTD here"]>
```

4 : تشکیل employees.xsd حاوی تعریف XML Schema دارد.

فایل xsd ایجاد شده ، تصویر زیر نمایی از بخشی از کدهای این فایل است.

```
employees.xsd
1 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
2   <xs:element name="employees">
3     <xs:complexType>
4       <xs:sequence>
5         <xs:element name="employee" maxOccurs="unbounded" minOccurs="0">
6           <xs:complexType>
7             <xs:sequence>
8               <xs:element name="name">
9                 <xs:complexType>
10                  <xs:sequence>
11                    <xs:element type="xs:string" name="lastname"/>
12                    <xs:element type="xs:string" name="firstname"/>
13                  </xs:sequence>
14                </xs:complexType>
15              </xs:element>
16              <xs:element type="xs:float" name="age" minOccurs="0"/>
17              <xs:element type="xs:string" name="phone" maxOccurs="unbounded"/>
18              <xs:element type="xs:date" name="hiredate"/>
19            </xs:sequence>
20          </xs:complexType>
21        </xs:element>
22      </xs:sequence>
23    </xs:complexType>
24  </xs:element>
25 </xs:schema>
```

یک فایل XSD محتویات خود را به صورت متن ساده در فرمت XML ذخیره می کند، به این معنی که فایل ها را می توان توسط هر ویرایشگر متن و برنامه های متعدد دیگری باز کرد و مشاهده کرد.

عنصر **complexType** یک نوع پیچیده را تعریف می کند. یک عنصر نوع پیچیده یک عنصر XML است که شامل عناصر و/یا ویژگی های دیگر است.

عنصر **sequence** مشخص می کند که عناصر فرزند باید در یک دنباله ظاهر شوند. هر عنصر فرزند می تواند از 0 تا چند بار رخ دهد.

همانطور که در لاین هفتم میبینید ، دیتا ها و تگ های تکرار شونده employee درون sequence تعریف شده اند ، همچنین این کار برای firstname و lastname که درون name قرار میگیرند تکرار شده است.

The XML document is valid.

Close

XML Input

Option 1: Copy-paste your XML document here

```
<?xml version="1.0" encoding="UTF-8"?>

<employees>
  <employee code="e1">
    <name>
      <lastname>Losa</lastname>
```

Option 2: Or upload your XML document

Choose File No file chosen

UTF-8

XSD Input (Optional if XSD referred in XML using schemaLocation)

Option 1: Copy-paste your XSD document here

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="employees">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="employee" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
```

Option 2: Or upload your XSD document

این اعتبار سنجی پیش از اعمال محدودیت بر روی price و اعمال بازه 20 تا 3000 است. نتیجه اعتبار سنجی به کمک وبسایت <https://www.freeformatter.com/xml-validator-xsd.html> را در تصویر بالا مشاهده میکنید.

ویژگی minOccurs حداقل تعداد دفعاتی را که عنصر می تواند رخ دهد را مشخص می کند.

ویژگی maxOccurs حداکثر تعداد دفعاتی را که عنصر می تواند رخ دهد را مشخص می کند.

می تواند مقدار هر عدد صحیح مثبتی بزرگتر یا مساوی با مقدار ویژگی minOccurs داشته باشد.

XML schema (XSD) data

```
1 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/
2         <xs:simpleType name="priceRestrict">
3             <xs:restriction base="xs:float">
4                 <xs:minExclusive value="20"/>
5                 <xs:maxExclusive value="3000"/>
6             </xs:restriction>
7         </xs:simpleType>
8     <xs:element name="employees">
9         <xs:complexType>
10             <xs:sequence>
11                 <xs:element name="employee" maxOccurs="unbounded" minOccurs="0">
12                     <xs:complexType>
13
```

Validate

Document Valid

اعتبار سنجی پس از اعمال محدودیت بر روی price و اعمال بازه 20 تا 3000 است. این تست به کمک وبسایت <https://www.liquid-technologies.com/online-xsd-validator> انجام شده است. همانطور که در تصویر مشخص است یک تایپ جدید به نام priceRestrict ایجاد کرده ایم و در قسمت تعریف price ، تایپ آن را برابر با این تایپ قرار می‌دهیم.

```
<xs:element name="price" maxOccurs="1" minOccurs="1" type="priceRestrict" />
```

5 : تشکیل employees_xsd.xml که اعلان اتصال به XML Schema دارد.
به سادگی به کمک قطعه کد زیر این اتفاق رخ میدهد. قطعه کد زیر درون فایل XML قرار میگیرد.

```
<?xml version="1.0" encoding="UTF-8"?>

<employees
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="employees.xsd">
  <employee code="e1">
```

6 : تشکیل xpath.txt حاوی عبارات XPath.

/employees/employee[@code='e1' or @code='e3']/name/firstname

/employees/employee[@code='e1' or @code='e3']/name/firstname**Elements found: 2**

XML modeFormatSave

```
:?xml version="1.0" encoding="UTF-8"?>

:employees>
<employee code="e1">
  <name>
    <lastname>Losa</lastname>
    <firstname>Loretta</firstname>
  </name>
  <phone>
    713-868-5455
  </phone>
  <phone>
    713-868-5455
  </phone>
```

1. Loretta
2. Simin

/employees/employee/devices/device[price > 150]

/employees/employee/devices/device[price > 150]

Elements found: 3

XML modeFormatSave

```
<?xml version="1.0" encoding="UTF-8"?>

employees>
employee code="e1">
  <name>
    <lastname>Losa</lastname>
    <firstname>Loretta</firstname>
  </name>
  <phone>
    713-868-5455
  </phone>
  <phone>
    713-868-5466
  </phone>
  <hiredate>2009-05-04</hiredate>
  <devices>
    <device type="t2">
      <item id="p111">Printer</item>
      <price >111.00</price>
    </device>
    <device type="t1">
      <item id="p222">Laptop</item>
      <price>989.00</price>
    </device>
  </devices>
</employee>
</employees>
```

1. Laptop 989.00

2. Desktop 2995.00

3. Scanner 200.00

<http://xpather.com> وبسایت تست

از وبسایت های زیبا ساز برای مرتب کردن کدها استفاده شده است.

