

# eXtensible Markup Language

```
<?xml version="1.0"?>
<Zamowienie data="2005-05-14">
  <Numer>123456</Numer>
  <Opis>
    Zamowienie.
  </Opis>
  <Produkty>
    <Produkt>
      <Ilosc jednostka="pudelko">1</Ilosc>
      <Nazwa>Płyta</Nazwa>
    </Produkt>
    <Produkt>
      <Ilosc jednostka="sztuka">45</Ilosc>
      <Nazwa>Dyskietka</Nazwa>
    </Produkt>
  </Produkty>
  <Klient>
    <IN>Zbigniew Nowak</IN>
    <Adres>Polna 14 Białystok</Adres>
  </Klient>
</Zamowienie>
```

# Poprawny dokument XML

## 1. WELL-FORMED

- Wszystkie znaczniki są zamknięte
- Wszystkie znaczniki są poprawnie zagnieżdżone
- Dokument posiada jeden znacznik “korzeń”
- Wartości atrybutów zamknięte w cudzysłowach
- Znaki specjalne ( < > “ ) reprezentowane jako “character entities”

## 2. VALID

- Składnia zgodna z pewną specyfikacją

## Parsery XML

```
graph TD; A[Parsery XML] --> B[sterowane zdarzeniami  
Simple API for XML]; A --> C[budujące drzewo  
Document Object Model];
```

*sterowane zdarzeniami*  
Simple API for XML

*budujące drzewo*  
Document Object Model

# Elementy dokumentu

- **Prolog**

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

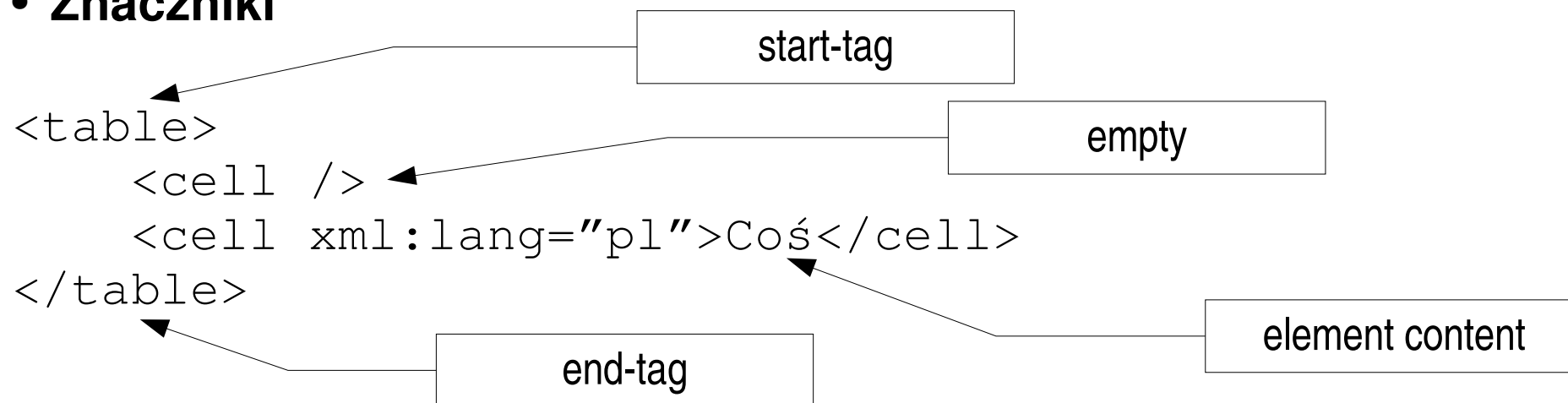
- **Document Type Definition**

```
<!DOCTYPE Rezerwacja [  
  <!ELEMENT Rezerwacja(#PCDATA)>  
>
```

- **Specyfikacja języka naturalnego**

```
<table>  
  <cell xml:lang="en">Something</cell>  
  <cell xml:lang="pl">Coś</cell>  
</table>
```

- **Znaczniki**



## Elementy dokumentu cd.

- **Entities (encje?)**

- parsed (alias) vs unparsed (zawartość)
- character ( *&#x3C;* ) vs general ( *&mail;* )
- predefiniowane ( *&lt;* *&gt;* *&quot;* )
- entity-reference ( *&ent;* ) vs parameter-entity reference ( *%ent;* )
- wewnętrzne vs zewnętrzne

- **Tekst**

- **sekcja CDATA**

- ```
<[CDATA[
```

- ```
Tutaj może być wszystko ... <> "fddfdf"
```

- ```
]]>
```

# Document Type Declaration

- **Korzeń**

```
<!DOCTYPE Zamowienie [  
    ...  
>
```

- **Element ( znacznik )**

```
<!DOCTYPE Zamowienie [  
    <!ELEMENT Zamowienie (#PCDATA)>  
>
```

`<!ELEMENT nazwa specyfikacja_zawartości>`

Zawartość:

- 'EMPTY'
- 'ANY'
- Mixed ( #PCDATA, sekwencja, wybór)
- children (sekwencja, wybór)

Liczebność: +, \*, ?

# Document Type Declaration cd.

- **Lista atrybutów**

```
<!ATTLIST termdef
      id          ID          #REQUIRED
      name       CDATA      #IMPLIED>
```

*<!ATTLIST nazwa\_elementu definicja\* >*  
*definicja = nazwa typ wartość\_domyślna*

Typy atrybutów:

- CDATA - StringType
- TokenizedType (obiekty, identyfikatory)
- EnumeratedType - lista wartości

Wartości domyślne:

- #REQUIRED
- #IMPLIED
- wartość ( z opcją #FIXED )

# Document Type Declaration cd.2

- **Sekcje warunkowe**

```
<![INCLUDE [ .. ]]>
```

```
<![IGNORE [ .. ]]>
```

- **Encje**

```
<!ENTITY mail "admin@ii.pb.bialystok.pl">
```

```
<!ENTITY nazwa definicja >  
<!ENTITY % nazwa definicja >  
<!ENTITY SYSTEM uri >  
<!ENTITY PUBLIC id_lokalne uri >
```

```
<!ENTITY lt "&#38;#60;">
```

```
<!ENTITY gt "&#62;">
```

```
<!ENTITY amp "&#38;#38;">
```

```
<!ENTITY apos "&#39;">
```

```
<!ENTITY quot "&#34;">
```

# XML Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="Zamowienie" type="ZamowienieType"/>
  <xsd:complexType name="ZamowienieType">
    <xsd:sequence>
      <xsd:element name="Numer" type="xsd:integer"/>
      <xsd:element name="Opis" type="xsd:string"/>
      <xsd:element name="Produkty" type="ProduktyType"/>
      <xsd:element name="Klient" type="KlientType"/>
      <xsd:element name="Realizacja" type="RType"/>
    </xsd:sequence>
    <xsd:attribute name="date" type="xsd:date"/>
  </xsd:complexType>
  <xsd:complexType name="ProduktyType">
    <xsd:sequence>
      <xsd:element name="Produkt" type="PType" minOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="PType">
    <xsd:sequence>
      <xsd:element name="Ilosc" type="xsd:integer" />
      <xsd:element name="Nazwa" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```