

NUMBER SYSTEM

Number systems are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

BINARY NUMBER SYSTEM

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

OCTAL NUMBER SYSTEM

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

DECIMAL NUMBER SYSTEM

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

HEXADECIMAL NUMBER SYSTEM

A Hexadecimal number system has sixteen (16) alphanumeric values from **0 to 9 and A to F**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here A is 10, B is 11, C is 12, D is 14, E is 15 and F is 16.

Number system	Base(Radix)	Used digits	Example
Binary	2	0,1	(11110000) ₂
Octal	8	0,1,2,3,4,5,6,7	(360) ₈
Decimal	10	0,1,2,3,4,5,6,7,8,9	(240) ₁₀
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	(F0) ₁₆

CONVERSIONS

DECIMAL TO OTHER

1. DECIMAL TO BINARY

Decimal Number System to Other Base

To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:

A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).

B) Write the remainder from step 1 as a Least Signification Bit (LSB) to Step last as a Most Significant Bit (MSB).

Decimal to Binary Conversion		Result	
Decimal Number is : (12345)₁₀		Binary Number is (11000000111001)₂	
2	12345		1 LSB
2	6172		0
2	3086		0
2	1543		1
2	771		1
2	385		1
2	192		0
2	96		0
2	48		0
2	24		0
2	12		0
2	6		0
2	3		1
1	1	1 MSB	

2. DECIMAL TO OCTAL

Decimal to Octal Conversion		Result	
Decimal Number is : (12345)₁₀		Octal Number is (30071)₈	
8	12345		1 LSB
8	1543		7
8	192		0
8	24		0
	3	3 MSB	

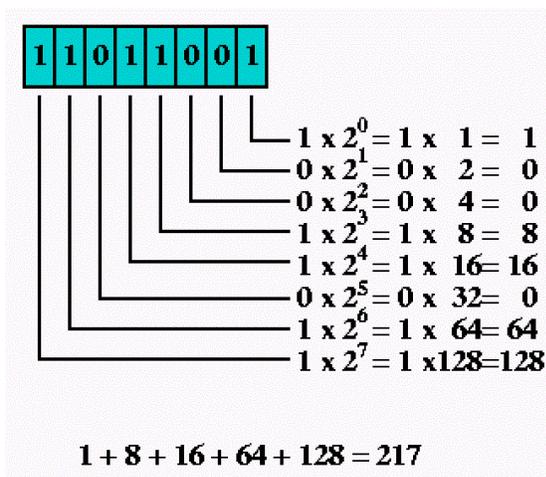
3. DECIMAL TO HEXADECIMAL

Decimal to Hexadecimal Conversion	Result												
<p>Example 1 Decimal Number is : $(12345)_{10}$</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>16</td><td>12345</td></tr> <tr><td>16</td><td>771</td></tr> <tr><td>16</td><td>48</td></tr> <tr><td>8</td><td>3</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>9</td></tr> <tr><td>3</td></tr> <tr><td>0</td></tr> <tr><td>3</td></tr> </table> <p>LSB</p> <p>MSB</p>	16	12345	16	771	16	48	8	3	9	3	0	3	Hexadecimal Number is $(3039)_{16}$
16	12345												
16	771												
16	48												
8	3												
9													
3													
0													
3													
<p>Example 2 Decimal Number is : $(725)_{10}$</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>16</td><td>725</td></tr> <tr><td>16</td><td>45</td></tr> <tr><td></td><td>2</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>5</td><td>5</td></tr> <tr><td>13</td><td>D</td></tr> <tr><td>2</td><td>2</td></tr> </table> <p>LSB</p> <p>MSB</p>	16	725	16	45		2	5	5	13	D	2	2	Hexadecimal Number is $(2D5)_{16}$ Convert 10, 11, 12, 13, 14, 15 to its equivalent... A, B, C, D, E, F
16	725												
16	45												
	2												
5	5												
13	D												
2	2												

BINARY TO OTHER

A) Multiply the digit with 2(with place value exponent). Eventually add all the multiplication becomes the Decimal number.

1. BINARY TO DECIMAL



2. BINARY TO OCTAL

An easy way to convert from binary to octal is to group binary digits into sets of three, starting with the least significant (rightmost) digits.

Binary: 11100101 =	11 100 101	
	011 100 101	Pad the most significant digits with zeros if necessary to complete a group of three.

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111
---------	-----	-----	-----	-----	-----	-----	-----	-----

Octal:	0	1	2	3	4	5	6	7
--------	---	---	---	---	---	---	---	---

Binary =	011	100	101	
Octal =	3	4	5	= 345 oct

3. BINARY TO HEXADECIMAL

An equally easy way to convert from binary to hexadecimal is to group binary digits into sets of four, starting with the least significant (rightmost) digits.

Binary: 11100101 = 1110 0101

Then, look up each group in a table:

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

OCTAL TO OTHER

1. OCTAL TO BINARY

Converting from octal to binary is as easy as converting from binary to octal. Simply look up each octal digit to obtain the equivalent group of three binary digits.

Octal:	0	1	2	3	4	5	6	7
Binary:	000	001	010	011	100	101	110	111

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

2. OCTAL TO HEXADECIMAL

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal. For example, to convert 345 octal into hex:

(from the previous example)

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):

Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7

Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

Therefore, through a two-step conversion process, octal 345 equals binary 011100101 equals hexadecimal E5.

3. OCTAL TO DECIMAL

The conversion can also be performed in the conventional mathematical way, by showing each digit place as an increasing power of 8.

$$345 \text{ octal} = (3 * 8^2) + (4 * 8^1) + (5 * 8^0) = (3 * 64) + (4 * 8) + (5 * 1) = 229 \text{ decimal}$$

OR

Converting octal to decimal can be done with repeated division.

1. Start the decimal result at 0.
2. Remove the most significant octal digit (leftmost) and add it to the result.
3. If all octal digits have been removed, you're done. Stop.
4. Otherwise, multiply the result by 8.
5. Go to step 2.

Octal Digits	Operation	Decimal Result	Operation	Decimal Result
345	+3	3	× 8	24
45	+4	28	× 8	224
5	+5	229	done.	

$$\Rightarrow (345)_8 = (229)_{10}$$

HEXADECIMAL TO OTHER

1. HEXADECIMAL TO BINARY

Converting from hexadecimal to binary is as easy as converting from binary to hexadecimal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

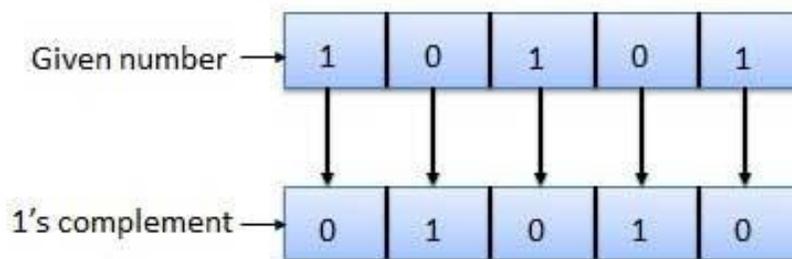
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	8	9	A	B	C	D	E	F
Binary:	1000	1001	1010	1011	1100	1101	1110	1111

Hexadecimal =	A	2	D	E	
Binary =	1010	0010	1101	1110	= 1010001011011110 binary

2. HEXADECIMAL TO OCTAL

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.



Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.

Case	A	+	B	Sum	Carry
1	0	+	0	0	0
2	0	+	1	1	0
3	1	+	0	1	0
4	1	+	1	0	1

In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example – Addition

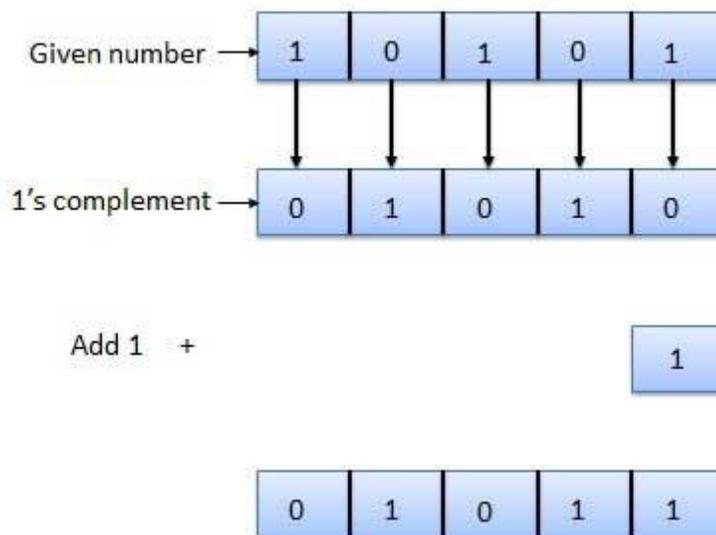
$$\begin{array}{r}
 0011010 + 001100 = 00100110 \\
 \begin{array}{r}
 11 \quad \text{carry} \\
 0011010 = 26_{10} \\
 +0001100 = 12_{10} \\
 \hline
 0100110 = 38_{10}
 \end{array}
 \end{array}$$

2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

$$2's \text{ complement} = 1's \text{ complement} + 1$$

Example of 2's Complement is as follows.



Rules of Binary Addition

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$, and carry 1 to the next more significant bit

For example,

$$\begin{array}{r}
 00011010 + 00001100 = 00100110 \\
 \begin{array}{r}
 \\
 + 0 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 = 26_{(\text{base } 10)} \\
 = 12_{(\text{base } 10)} \\
 = 38_{(\text{base } 10)}
 \end{array}$$

$$\begin{array}{r}
 00010011 + 00111110 = 01010001 \\
 \begin{array}{r}
 \\
 + 0 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 = 19_{(\text{base } 10)} \\
 = 62_{(\text{base } 10)} \\
 = 81_{(\text{base } 10)}
 \end{array}$$

Rules of Binary Multiplication

- $0 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 0 = 0$
- $1 \times 1 = 1$, and no carry or borrow bits

For example,

$$\begin{array}{r}
 00101001 \times 00000110 = 11110110 \\
 \begin{array}{r}
 \\
 \times 0 \\
 \hline
 \\
 \\
 0 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 = 41_{(\text{base } 10)} \\
 = 6_{(\text{base } 10)} \\
 = 246_{(\text{base } 10)}
 \end{array}$$

Binary Division

Binary division is the repeated process of subtraction, just as in decimal division.

For example,

$$00101010 \div 00000110 = 111 = 7_{(\text{base } 10)}$$

00000111

$$\begin{array}{r}
 110 \overline{) 00101010} = 42_{(base\ 10)} \\
 \underline{- 110} \\
 1001 \\
 \underline{- 110} \\
 110 \\
 \underline{- 110} \\
 0
 \end{array}$$

borrows

10000111 ÷ 00000101 =
00011011

$$\begin{array}{r}
 11011 \overline{) 1001111} = 27_{(base\ 10)} \\
 \underline{- 101} \\
 1001 \\
 \underline{- 101} \\
 111 \\
 \underline{- 101} \\
 101 \\
 \underline{- 101} \\
 0
 \end{array}$$

Example – Division

101010 / 000110 = 000111

$$\begin{array}{r}
 111 = 7_{10} \\
 000110 \overline{) 101010} = 42_{10} \\
 \underline{- 110} = 6_{10} \\
 1001 \\
 \underline{- 110} \\
 110 \\
 \underline{- 110} \\
 0
 \end{array}$$

HTML

Tags are **containers**. The *html tag* indicates that everything between `<html>` and `</html>` is code that conforms to the standards of the type of HTML. Everything between the opening tag and closing tag are inside that tag and therefore have the attributes that tag gives them. Those attributes can be modified.

SR. NO	TAG NAME	ATTRIBUTES	DESCRIPTION	EXAMPLE
1.	<HEAD>	The following elements can go inside the <head> element: <title>	The <head> element is a container for all the head elements. The <head> element can include a title for the document, scripts, styles, meta information, and more.	<pre><!DOCTYPE html> <html> <head> <title>Title of the document</title> </head> <body> The content of the document..... </body></html></pre>
2.	<A>	href target=_blank _parent _self _top	The <a> tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link's destination.	<pre><!DOCTYPE html> <html> <head> <title>Title of the document</title> </head> <body> Visit yahoo.com! </body></html></pre>
3.	<P>	align	The <p> tag defines a paragraph.	<pre><!DOCTYPE html> <html> <head><title>Title of the document</title> </head> <body> <p>This is some text in a paragraph.</p> </body></html></pre>
4.		color face size	The tag specifies the font face, font size, and color of text.	<pre><html> <head><title>Title of the document</title> </head> <body>This is some text! This is some text! This is some text! </body></html></pre>
5.	<H1> - <H6>	align	The <h1> to <h6> tags are used to define HTML headings. <h1> defines the most important heading. <h6> defines the least important heading.	<pre><html> <head><title>Title of the document</title> </head> <body> <h1>This is heading 1</h1> <h2>This is heading 2</h2> <h3>This is heading 3</h3></pre>

				<pre><h4>This is heading 4</h4> <h5>This is heading 5</h5> <h6>This is heading 6</h6> </body></html></pre>
6.	<TABLE>	align bgcolor border cellpadding ng cellspacing g width	<p>The <table> tag defines an HTML table.</p> <p>An HTML table consists of the <table> element and one or more <tr>, <th>, and <td> elements.</p> <p>The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.</p>	<pre><html> <head><title>Title of the document</title> </head> <body> <table> <tr> <th>Month</th> <th>Savings</th> </tr> <tr> <td>January</td> <td>\$100</td> </tr> </table> </body></html></pre>
7.		compact type	<p>The tag defines an unordered (bulleted) list.</p> <p>Use the tag together with the tag to create unordered lists.</p>	<pre><html><head><title>Title of the document</title> </head><body> Coffee Tea Milk </body></html></pre>
8.		align alt border height src width	<p>The tag defines an image in an HTML page.</p> <p>The tag has two required attributes: src and alt.</p>	<pre><html> <head><title>Title of the document</title> </head> <body>< </body> </html></pre>
9.	<HR>	align size width noshade	<p>The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic).</p> <p>The <hr> element is used to separate content (or define a change) in an HTML page.</p>	<pre><h1>HTML</h1> <p>HTML is a language for describing web pages.....</p> <hr> <h1>CSS</h1> <p>CSS defines how to display HTML elements.....</p></pre>
10.	<VIDEO>	autoplay height loop src width controls	<p>The <video> tag specifies video, such as a movie clip or other video streams.</p> <p>Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg.</p>	<pre><html> <head><title>Title of the document</title> </head> <body><video width="320" height="240" controls> <source src="movie.mp4" type="video/mp4"> <source src="movie.ogg" type="video/ogg"> Your browser does not support the video tag. </video> </body> </html></pre>
11.	<LINK>	href rel type	<p>The <link> tag defines a link between a document and an external resource. The <link> tag is used to link to external style sheets.</p>	<pre><head> <link rel="stylesheet" type="text/css" href="theme.css"> </head></pre>

XML

XML stands for **Extensible Markup Language** and is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions:

- **XML is extensible:** XML allows you to create your own self-descriptive tags, or language, that suits your application.
- **XML carries the data, does not present it:** XML allows you to store the data irrespective of how it will be presented.
- **XML is a public standard:** XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Usage

A short list of XML usage says it all:

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable. So *what exactly is a markup language?* Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. More specifically, a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

Following example shows how XML markup looks, when embedded in a piece of text:

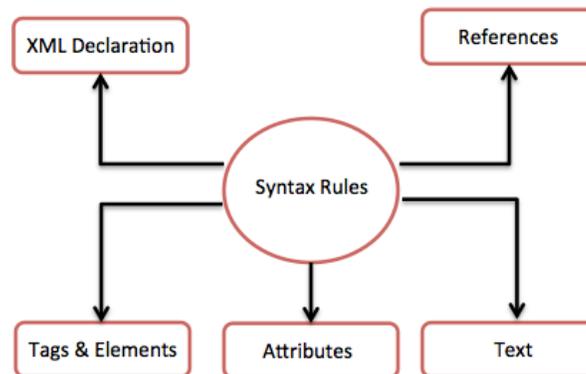
```
<message>  
<text>Hello, world!</text>  
</message>
```

This snippet includes the markup symbols, or the tags such as `<message>...</message>` and `<text>... </text>`. The tags `<message>` and `</message>` mark the start and the end of the XML code fragment. The tags `<text>` and `</text>` surround the text Hello, world!.

Example:

```
<?xml version="1.0"?>
<contact-info>
<name>Tanmay Patil</name>
<company>TutorialsPoint</company>
<phone>(011) 123-4567</phone>
</contact-info>
```

The following diagram depicts the syntax rules to write different types of markup and text in an XML document.



Each component of the above diagram in detail:

XML Declaration

The XML document can optionally have an XML declaration. It is written as below:

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

Where *version* is the XML version and *encoding* specifies the character encoding used in the document.

Syntax Rules for XML declaration

- The XML declaration is case sensitive and must begin with "`<?xml>`" where "xml" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs to be the first statement in the XML document.

- An HTTP protocol can override the value of *encoding* that you put in the XML declaration.

Tags and Elements

An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. XML-elements' names are enclosed by triangular brackets < > as shown below:

```
<element>
```

Syntax Rules for Tags and Elements

Element Syntax: Each XML-element needs to be closed either with start or with end elements as shown below:

```
<element>...</element>
```

or in simple-cases, just this way:

```
<element/>
```

CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** – CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.

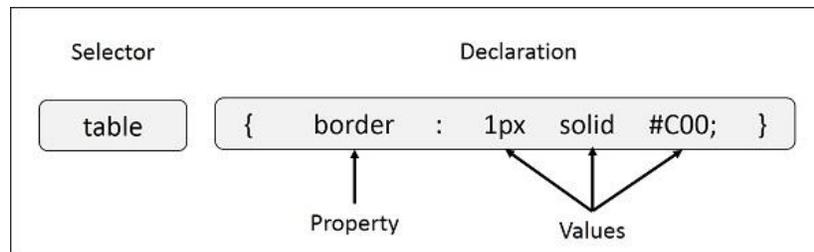
- **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

- **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** - Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows –

```
selector { property: value }
```



Example: You can define a table border as follows –

```
table{ border :1px solid #C00; }
```

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings:

```
h1 {
  color: #36CFFF;
}
```

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

```
* {  
  color: #000000;  
}
```

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to `` element only when it lies inside `` tag.

```
ul em {  
  color: #000000;  
}
```

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {  
  color: #000000;  
}
```

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example:

```
h1.black {  
  color: #000000;  
}
```

This rule renders the content in black for only `<h1>` elements with class attribute set to *black*.

You can apply more than one class selectors to given element. Consider the following example:

```
<p class="center bold">  
  This para will be styled by the classes center and bold.  
</p>
```

The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

```
#black {  
  color: #000000;  
}
```

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example –

```
h1#black {  
  color: #000000;  
}
```

This rule renders the content in black for only `<h1>` elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors, For example:

```
#black h2 {  
  color: #000000;  
}
```

In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having *id* attribute set to *black*.

Example:

```
<!DOCTYPE html>  
<html>  
  <head>  
  
    <style type = "text/css" media = "all">  
      body {  
        background-color: linen;  
      }  
      h1 {  
        color: maroon;  
        margin-left: 40px;
```

```
    }
  </style>

</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Inline CSS - The *style* Attribute

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax –

```
<element style = "...style rules....">
```

Attributes

Attribute	Value	Description
style	style rules	The value of <i>style</i> attribute is a combination of style declarations separated by semicolon (;).

Example

Following is the example of inline CSS based on the above syntax –

```
<html>
  <head>
  </head>
  <body>
    <h1 style = "color:#36C;"> This is inline CSS </h1>
  </body>
</html>
```

External CSS - The <link> Element

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file –

```
<head>
  <link type = "text/css" href = "... " media = "... " />
</head>
```

Example

Consider a simple style sheet file with a name *mystyle.css* having the following rules –

```
h1, h2, h3 {
  color: #36C;
  font-weight: normal;
  letter-spacing: .4em;
  margin-bottom: 1em;
  text-transform: lowercase;
}
```

Now you can include this file *mystyle.css* in any HTML document as follows –

```
<head>
  <link type = "text/css" href = "mystyle.css" media = " all" />
</head>
```

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well.

The process is an example of the client/server model. All computers that host Web sites must have Web server programs. Leading Web servers include Apache (the most widely-installed Web server), Microsoft's Internet Information Server (IIS) and nginx (pronounced *engine X*) from NGINX. Other Web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.

Web servers often come as part of a larger package of Internet- and intranet-related programs for serving email, downloading requests for File Transfer Protocol ([FTP](#)) files, and building and publishing Web pages. Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it.

There are four leading web servers – Apache, IIS, lighttpd and Jigsaw. Apart from these Web Servers, there are other Web Servers also available in the market but they are very expensive. Major ones are Netscape's iPlanet, Bea's Web Logic and IBM's WebSphere.



Apache HTTP Server

This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, Unix, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.



Internet Information Services

The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.



lighttpd

The **lighttpd**, pronounced *lighty* is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.



Sun Java System Web Server

This web server from Sun Microsystems is suited for medium and large websites. Though the server is free it is not open source. It however, runs on Windows, Linux and Unix platforms.

The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, Ruby on Rails, ASP and Coldfusion etc.



Jigsaw Server

Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, Unix, Windows, Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

DATABASE

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational DataBase Management System (RDBMS)** is a software that:

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

RDBMS TERMINOLOGY:

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

- **Database:** A database is a collection of tables, with related data.
- **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.

- **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy:** Storing data twice, redundantly to make the system faster.
- **Primary Key:** A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.
- **Foreign Key:** A foreign key is the linking pin between two tables.
- **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index:** An index in a database resembles an index at the back of a book.
- **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

MS ACCESS

Microsoft Access is a unique tool released by Microsoft that provides both the functionality of a database and the programming capabilities to create end-user screens. Each version of Access can "look and feel" completely different from another.

Microsoft Access stores information in what is called a database. Microsoft Access is an information management tool that helps you store information for reference, reporting, and analysis. Microsoft Access helps you analyze large amounts of information, and manage related data more efficiently than Microsoft Excel or other spreadsheet applications.

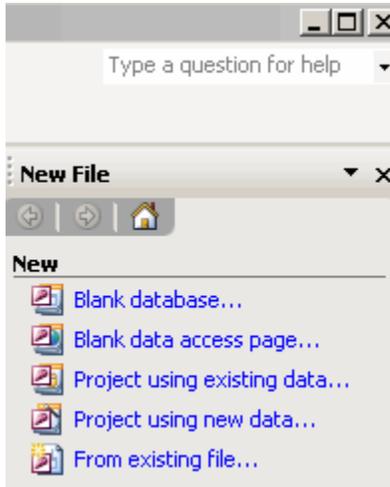
There are four major steps to using Microsoft Access:

1. **Database Creation:** Create your Microsoft Access database and specify what kind of data you will be storing. A retail business might create a database to store all their sales information (i.e. items sold, customer, employee, commission, etc)
2. **Data Input:** After your database is created the data the store gathers every business day can be entered into the Access database.
3. **Query:** This is a fancy term to basically describe the process of retrieving information from the database.
4. **Report(optional):** Information from the database is organized in a nice presentation that can be printed in an Access Report.

CREATING ACCESS DATABASE

To create a new database in Access follow these steps:

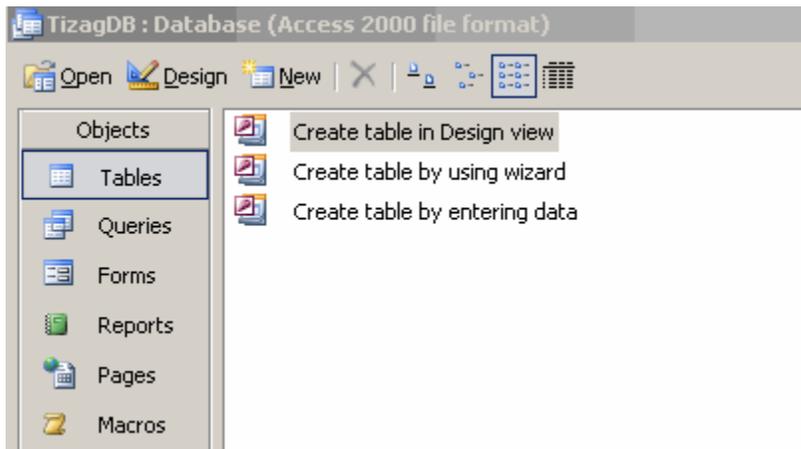
1. Start Access
2. From the menu choose File < New
3. The "New File" side bar will be displayed on the right-hand side of the screen



4. Left-click the option "Blank database...", which will then ask you to name your database. Helpful Hint: Access databases are saved with the .mdb extension.
5. Name your file (we chose TizagDB) and press Create. This will automatically save your blank database, so remember where you put it!



6. The Access Database interface should now be displayed and you are well on your way to learning Access!



Now that the database has been created we can begin to create our first Access Table. A table resides within a database and holds information specific to a certain area.

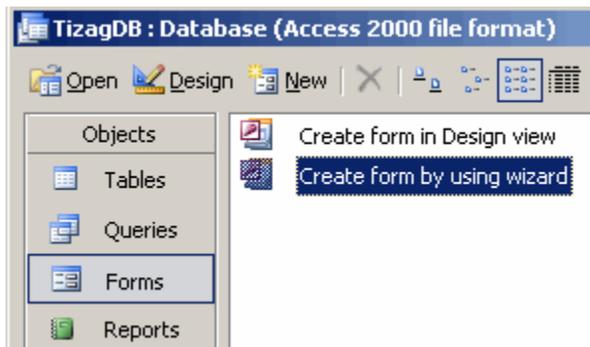
ACCESS FORMS

Access provides an easy way to enter data into your Access tables with forms. In Access you have the ability to quickly make and customize these data entry forms to streamline the data input process.

CREATING AN ACCESS FORM

Although we haven't recommended the various wizards that Access had available in the previous lessons, the **form wizard** is actually very useful and should save you a bunch of time! Let's create a simple data input form for the new employee!

1. Navigate to the **Forms** section in Access



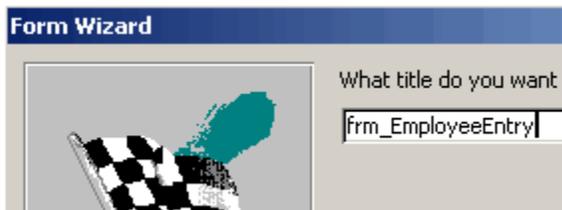
2. Double-click "Create form by using wizard"
3. We want all of the fields from **tbl_Sales** to be included in this form, so first select **tbl_Sales** from the drop down box "Tables/Queries"



4. The single right arrow will add one selected field at a time, but we want all the fields. The shortcut to add every field from a given table or query is to click the double arrow button. Do that and click **Next**.

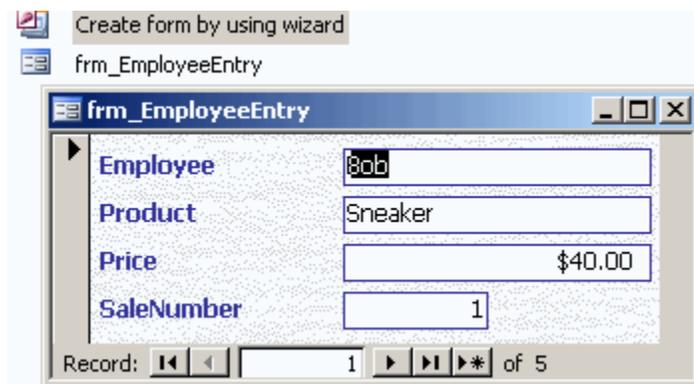


5. Choose a **Columnar** layout and press **Next**
6. Choose any style and press **Next** (we chose "Sumi Painting")
7. Change the form's title to frm_EmployeeEntry



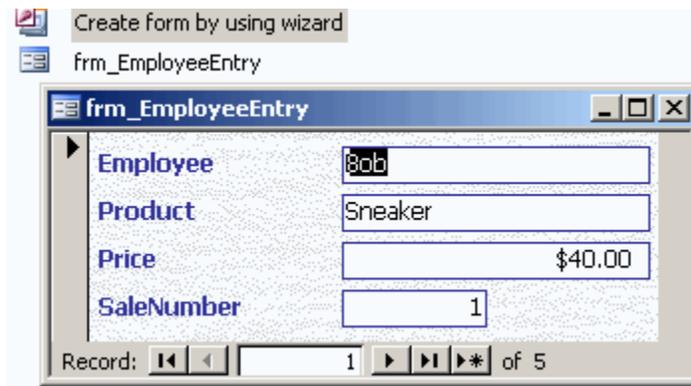
and click **Finish**

Open up your form and check it out!



ENTERING DATA USING ACCESS FORMS

1. Open up **frm_EmployeeEntry** (easy!)



The screenshot shows a Microsoft Access form titled "frm_EmployeeEntry". The form has a title bar with the text "frm_EmployeeEntry" and standard window controls. Below the title bar, there are four data entry fields:

Employee	Bob
Product	Sneaker
Price	\$40.00
SaleNumber	1

At the bottom of the form, there is a record navigation bar with the text "Record: 1 of 5" and several navigation arrows.

2. At the bottom of the form is a set of arrows to navigate through the records. To get to the end of the existing records and begin entering data you need to click the arrow with an asterisk(*).



3. Clicking that button will bring you to the first blank record, which would be the **sixth** in our case. You would then enter all the data for that record and click the right arrow to advance to the next blank record. After all the new records have been entered, close the form and pat yourself on the back.

When you enter data into this form it will automatically add it to our existing **tbl_Sales** because we specified that table when we created our form. With this form the new employee will be able to enter data into Bob's existing Access table without ever knowing a thing about tables!

MYSQL

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

DIFFERENCES BETWEEN MS ACCESS & MYSQL

MySQL is a database. With it you get tables, views (queries), and stored procedures. You have to write the user interface with something else. That could be web-based, Visual Basic, pretty much anything.

MS Access is a database (tables, queries), plus a built-in way to create the user interface (forms, reports and a programming language (VBA) to support the whole thing).

The database in MS Access is limited to 2 gigabytes. MySQL is much larger.

Generally, if you are going to have a single developer with a few users, MS Access will allow you to produce an application quickly, particularly if you have experience with VB or VBA.

If you will have many users and several developers you will be better off with a dedicated database (like MySQL) and application front-ends as appropriate.