April 2018

1. What do you mean by “class” and “id” in CSS? Explain with an example? (3) -1

- The class attribute is used to specify a class for an element. Multiple elements can share the same class. To define CSS properties for a class the period (.) character followed by the name of the class is used and properties are defined in curly braces.

  <style>

  </style>

- The Id attribute specifies a unique id for an element. The value of id should be unique within the document.

  <style>

  #myheader(

  Background-color: blue;

  Color:red;

  Padding:10px;

  }

  .city{
  Background-color: tomato;
  Color: white;
  Padding: 5px;
  }
  </style>

  <h1 id=”myheader”>World Cities</h1>

  <h2 class=”city”> London </h2>
   London is in England
  <h2 class=”city”>NewYork </h2>
  NewYork is in the US

2. How to link an external style sheet to an HTML document? What are the attributes required for that? (3) -4

An external style sheet may be linked to an HTML document through HTML's **LINK** element:

Eg: `<link rel="stylesheet" href="styles.css">`

Attributes:

rel : Required. Specifies the relationship between the current document and the linked document

href: Specifies the location of the linked document

3. What is bootstrap? What is the advantage of using bootstrap? (4) -17

Bootstrap is a giant collection of handy, reusable bits of code written in HTML, CSS, and JavaScript. It's also a front-end development framework that enables developers & designers to quickly build fully responsive websites.

The following are the advantages of using Bootstrap:

- Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead of them in separate files.
- It is supported by all popular browsers.
- With just the knowledge of HTML and CSS, anyone can get started with Bootstrap.
- Bootstrap's responsive CSS adjusts to Desktops, Tablets, and Mobiles.

4. Write CSS code for the following: (5) -18

i) set the background color for the hover and active link states to "yellow".

ii) Set the list style for unordered lists to "square".

iii) Set "Flower.gif" as the background image of the page

iv) Set dashed border for the document.

i) a:hover, a:active {background-color:yellow }

ii) ul { list-style: square }

iii) body { background-image: url("flower.gif") }

iv) body { border: 1px dashed black }

5. Differentiate between and tags with examples. (4) -19

Differences between and tag:

<DIV>	<SPAN>
The tag is a block level element.	The tag is an inline element.
It is best to attach it to a section of a web page.	It is best to attach a CSS to a small section of a line in a web page.
It accepts align attribute.	It does not accept align attribute.
This tag should be used to wrap a section, for highligting that section.	This tag should be used to wrap any specific word that you want to highlight in your webpage.

Eg - !DOCTYPE html>
<htm>

<body>
 div tag
 div tag
span tag
span tag

</body
</html>
Output-
div tag
div tag
span tag span tag

6. Explain XML document structure? (5) -27

- XML documents are formed as element trees.
- An XML tree starts at a root element and branches from the root to child elements.
- All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

- The terms parent, child, and sibling are used to describe the relationships between elements.
- Parents have children. Children have parents. Siblings are children on the same level (brothers and sisters).
- All elements can have text content and attributes.
- XML uses a much self-describing syntax.
- A prolog defines the XML version and the character encoding:

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

- The next line is the root element of the document.
- The next line starts a child element.
- The child elements can have many child elements.

- Eg :

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

7. What is the requirement of XML Schema? Explain. (5) -34

In the XML world, hundreds of standardized XML formats are in daily use.Many of these XML standards are defined by XML Schemas.XML Schema is an XML-based (and more powerful) alternative to DTD.

DTDs have several disadvantages :- DTDs are written in a syntax unrelated to XML. DTDs do not allow restrictions on the form of data that can be the content of a particular tag.There are only 10 data types, none of which is numeric.

XML Schema standard was designed by W3C and has been developed to attempt to overcome the weakness of DTD. An XML schema is an XML document- can be parsed with an XML parser that provides far more control over data types. There are 44 different data types. Also, the user can define new types with constraints on existing data types.

The following usage scenarios describe XML applications that should benefit from XML schemas.

- Publishing and syndication
- Electronic commerce transaction processing.
- Supervisory control and data acquisition.
- Traditional document authoring/editing governed by schema constraints
- Use schema to help query formulation and optimization.
- Open and uniform transfer of data between applications, including databases.
- Metadata Interchange

8. Explain simple and complex data types in XML with suitable examples? (10) -38

   There are 2 categories of user-defined data types - simple and complex

   - Simple data type
     - This is a data type whose content is restricted to strings.
     - A simple type cannot have attributes or include nested elements.
     - A large collection of predefined data types is included in the category.
   - Complex data type
     - It can have attributes and include other data types as child elements.

The XML Schema defines 44 data types, 19 of which are primitive and 25 of which are derived. The primitive data types include - string, Boolean, float, time, and anyURI. The predefined derived types include - byte, long, decimal, unsignedInt, positiveInteger, and NMTOKEN.

User-defined data types are defined by specifying **restrictions** on an existing type. Then existing data type are called a **base type** and user-defined types are **derived types**

Constraints on derived types are given in terms of the facets of the base type. Eg: The integer primitive data type has eight possible facets: totalDigits, maxInclusive, maxExclusive, minInclusive, minExclusive, pattern, enumeration, and whitespace

Both simple and complex types can be **named or anonymous**. If anonymous, a type cannot be used outside the element in which it is declared. Context is essential to defining the meaning of a reference to an element in an XML schema.

Data declarations in an XML schema can be either **local or global.** A local declaration appears inside an element that is a child of the schema element. Global elements are visible in the whole schema in which they are declared.

Simple data types -

- Elements are defined in an XML schema with the **element** tag, which is from the XMLSchema namespace.
- Prefix xsd is normally used for names from this namespace
- **name attribute** gives the name of element
- **Type attribute**, which is used to specify the type of content allowed in the element
  Eg: <xsd:element name = “engine” type = “xsd:string” />

- An instance of the schema in which the engine element is defined could have the following element: <engine> inline six cylinder fuel injected </engine>
- An element can be given a default value with the default attribute:

Eg: <xsd:element name = “engine” type = “xsd:string” default = “fuel injected V-6” />

- Elements can have constant values defined using fixed attribute

Eg: <xsd:element name = “plane” type = “xsd:string” fixed=”single wing” />

- A simple user-defined data type is described in a simpleType element
- Facets must be specified in the content of a restriction element, which gives the base type name.
- The facets themselves are given in elements named for the facets: the value attribute specifies the value of the facet.
- user-defined type, firstName, for strings of fewer than 11 characters:

```
<xsd:simpleType name = "firstName">
  <xsd:restriction base = "xsd:string">
    <xsd:maxLength value = "10" />
  </xsd:restriction>
</xsd:simpleType>
```

- length facet is used to restrict the string to an exact number of characters
- minLength facet is used to specify a minimum length
- The number of digits of a decimal number can be restricted with the precision facet

```
<xsd:simpleType name = "phoneNumber">
  <xsd:restriction base = "xsd:decimal">
    <xsd:precision value = "7" />
  </xsd:restriction>
</xsd:simpleType>
```

Complex data types -

- defined with the **complexType** tag
- The elements that are the content of an element-only element must be contained in an ordered group, an unordered group, a choice, or a named group.
- The sequence element is used to contain an ordered group of elements

```
<xsd:complexType name = "sports_car">
  <xsd:sequence>
    <xsd:element name = "make"  type = "xsd:string" />
    <xsd:element name = "model"  type = "xsd:string" />
    <xsd:element name = "engine"  type = "xsd:string" />
    <xsd:element name = "year"  type = "xsd:decimal" />
  </xsd:sequence>
</xsd:complexType>
```

- A complex type whose elements are an unordered group is defined in an all element.

Example -xml schema

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

<!-- planes.xsd
     A simple schema for planes.xml
     -->
<xsd:schema
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  targetNamespace = "http://cs.uccs.edu/planeSchema"
  xmlns = "http://cs.uccs.edu/planeSchema"
  elementFormDefault = "qualified">

  <xsd:element name = "planes">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name = "make"
                     type = "xsd:string"
                     minOccurs = "1"
                     maxOccurs = "unbounded" />
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Xml document:

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

<!-- planes1.xml
     A simple XML document for illustrating a schema
     The schema is in planes.xsd
     -->
<planes
  xmlns = "http://cs.uccs.edu/planeSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://cs.uccs.edu/planeSchema
                        planes.xsd">
    <make> Cessna </make>
    <make> Piper </make>
    <make> Beechcraft </make>
</planes>
```

9. How XSLT style sheets can be used to control page layout in XML? (5) -41

10. Differentiate between JSON and XML? (5) -48

JSON	**XML**
JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON is supported by most browsers.	Cross-browser XML parsing can be tricky
JSON has no display capabilities.	XML offers the capability to display data because it is a markup language.
JSON supports only text and number data type.	XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data.
Retrieving value is easy	Retrieving value is difficult
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit

A fully automated way of deserializing/serializing JavaScript.	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object.	The object has to be express by conventions - mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding.
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.
It does not provide any support for namespaces.	It supports namespaces.
It is less secured.	It is more secure than JSON.

Examples
JSON

```
{"Geeks":[
   { "firstName":"Vivek", "lastName":"Kothari" },
   { "firstName":"Suraj", "lastName":"Kumar" }
  ]}
```

XML

```
<Geeks>
   <Geek>
      <firstName>Vivek</firstName> <lastName>Kothari</lastName>
   </Geek>
   <Geek>
      <firstName>Suraj</firstName> <lastName>Kumar</lastName>
   </Geek>
   </Geeks>
```

11. What are the different ways to create an array in PHP? Explain with examples (4) -51

In PHP, the array() function is used to create an array:

array();

There are basically three types of arrays in PHP:

- **Indexed or Numeric Arrays:** An array with a numeric index where values are stored linearly.

```
<?php

// One way to create an indexed array

$name_one = array("Zack", "Anthony", "Ram");

// Accessing the elements directly

echo "Accessing the 1st array elements directly:\n";

echo $name_one[2], "\n";

echo $name_one[0], "\n";

// Second way to create an indexed array

$name_two[0] = "ZACK";

$name_two[1] = "ANTHONY";

$name_two[2] = "RAM";

// Accessing the elements directly

echo "Accessing the 2nd array elements directly:\n";

echo $name_two[2], "\n";

echo $name_two[0], "\n";

?>
```

Output: Accessing the 1st array elements directly:

Ram

Zack

Accessing the 2nd array elements directly:

RAM

ZACK

- **Associative Arrays:** An array with a string index where instead of linear storage, each value can be assigned a specific key.

```
<?php

// One way to create an associative array

$name_one = array("Zack"=>"Zara", "Anthony"=>"Any",
"Ram"=>"Rani", "Salim"=>"Sara",  "Raghav"=>"Ravina");

// Second way to create an associative array

$name_two["zack"] = "zara";

$name_two["anthony"] = "any";

$name_two["ram"] = "rani";

$name_two["salim"] = "sara";

$name_two["raghav"] = "ravina";

// Accessing the elements directly

echo "Accessing the elements directly:\n";

echo $name_two["zack"], "\n";

echo $name_two["salim"], "\n";

echo $name_two["anthony"], "\n";

echo $name_one["Ram"], "\n";
```

```
echo $name_one["Raghav"], "\n";

?>
```

Output:

Accessing the elements directly:

zara

sara

any

Rani

Ravina

- **Multidimensional Arrays:** An array which contains a single or multiple array within it and can be accessed via multiple indices.

```
<?php


// Defining a multidimensional array
$favorites = array(
   array(
        "name" => "Dave Punk",
        "mob" => "5689741523",
        "email" => "davepunk@gmail.com",
   ),
   array(
        "name" => "Monty Smith",
```

```
        "mob" => "2584369721",
        "email" => "montysmith@gmail.com",
    ),
    array(
        "name" => "John Flinch",
        "mob" => "9875147536",
        "email" => "johnflinch@gmail.com",
    )
);

// Accessing elements
echo "Dave Punk email-id is: " . $favorites[0]["email"], "\n";
echo "John Flinch mobile number is: " . $favorites[2]["mob"];
?>
```

Output:

Dave Punk email-id is: davepunk@gmail.com

John Flinch mobile number is: 9875147536

12. What is server side scripting? Write a PHP program to check whether the given number is Armstrong or not? (6) -55

- **Server-side scripting** is a method of designing websites so that the process or user request is run on the originating server.
- Server-side scripts provide an interface to the user and are used to limit access to proprietary data and help keep control of the script source code.

- It involves employing scripts on a web server which produce a response customized for each user's (client's) request to the website. The alternative is for the web server itself to deliver a static web page.
- Scripts can be written in any of a number of server-side scripting languages.

- Server-side scripting is distinguished from client-side scripting where embedded scripts, such as JavaScript, are run client-side in a web browser, but both techniques are often used together.
- It is often used to provide a **customized interface for the user**.
- These scripts may assemble client characteristics for use in customizing the response based on those characteristics, the user's requirements, access rights, etc.
- Server-side scripting also enables the website owner to **hide the source code** that generates the interface, whereas with client-side scripting, the user has access to all the code received by the client.
- **Down-side**:
    - The client needs to make further requests over the network to the server in order to show new information to the user via the web browser.
    - These requests can slow down the experience for the user, place more load on the server, and prevent use of the application when the user is disconnected from the server.

PHP program to check Armstrong number:

```
<?php
// function to check whether the number is
// armstrong number or not
function armstrongCheck($number) {
	$sum = 0;
	$x = $number;

	// find the number of digits
	$numDigits = strlen((string)$x);

	while($x != 0)
	{
		$rem = $x % 10;
		$sum = $sum + pow($rem, $numDigits);
		$x = $x / 10;
	}

	// if true then armstrong number
	if ($number == $sum)
		return 1;
```

```
        // not an armstrong number
        return 0;
}

// Driver Code
$number = 407;
$flag = armstrongCheck($number);
if ($flag == 1)
        echo "Yes";
else
        echo "No"
?>
```

13. How can we declare attributes of an element in DTD? (5) -58

Basic syntax of DTD attributes declaration is as follows −

```
<!ATTLIST element-name attribute-name attribute-type
default-option>
```

In the above syntax −

- The DTD attributes start with <!ATTLIST keyword if the element contains the attribute.
- element-name specifies the name of the element to which the attribute applies.
- attribute-name specifies the name of the attribute which is included with the element-name.
- default option can specify either an actual value or a requirement for the value of the attribute

Eg:-

```
<?xml version = "1.0"?>
<!DOCTYPE address [
   <!ELEMENT address ( name )>
   <!ELEMENT name ( #PCDATA )>
   <!ATTLIST name id CDATA #REQUIRED>
]>

<address>
   <name id = "123">Tanmay Patil</name>
</address>
```

There are 10 different attribute types.

- CDATA is commonly used

Possible default options for attributes

Value	Meaning
A value	The quoted value, which is used if none is specified in an element
#FIXED value	The quoted value, which every element will have and which cannot be changed.
#REQUIRED	No default value is given; every instance of the element must specify a value
#IMPLIED	No default value is given (the browser chooses the default value); the value may or may not be specified in an element

Rules of Attribute Declaration

- All attributes used in an XML document must be declared in the Document Type Definition (DTD) using an Attribute-List Declaration
- Attributes may only appear in start or empty tags.
- The keyword ATTLIST must be in upper case
- No duplicate attribute names will be allowed within the attribute list for a given element.

14. Design an HTML form for entering a number by the user. Write a PHP code to display a message indicating, whether the number is odd or even, when clicking on the submit button. (5) -1

```
<html>
<body>
<form method="post">
   Enter a number:
   <input type="number" name="number">
   <input type="submit" value="Submit">
</form>
</body>
```

```
</html>
<?php
   if($_POST){
      $number = $_POST['number'];
      if(($number % 2) == 0){
         echo "$number is an Even number";
      }else{
         echo "$number is Odd number";
      }
   }
?>
```

15. Write a PHP script to count the instances of words in a string? (5) -4

```
<html>
<body>
<form method="post">
    Enter a string:
    <input type="text" name="text">
    <input type="submit" value="Submit">
</form>
</body>
</html>
<?php
    if($_POST){
        $string = $_POST[‘text];
$str_arr = explode(' ', $str);

foreach($str_arr as $word) {

    echo $word.": ".substr_count($string, $word)." <br />";
```

```
}

?>
```

16. What is the significance of cookies on the web? How can a cookie be created and destroyed in PHP? (5) -17

Cookies (also referred to as HTTP cookies or browser cookies) are small text files stored in a web user's browser directory or data folder.
Secure websites use cookies to validate a user's identity as they browse from page to page; without cookies, login credentials would have to be entered between before every product added to cart or wish list. Cookies enable and improve:
• Customer log-in
• Persistent shopping carts
• Wish lists
• Product recommendations
• Custom user interfaces (i.e. "Welcome back, Steve")
• Retaining customer address and payment information

Use **setcookie** to create a cookie with PHP which will expire after 30 days.
Example:

```
$cookie_name = 'pontikis_net_php_cookie';
$cookie_value = 'test_cookie_set_with_php';
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), '/'); // 86400 = 1 day
```

There is no direct way to delete a cookie. But we could use setcookie with expiration date in the past, to trigger the removal mechanism in our browser.
Example:

```
$cookie_name = 'pontikis_net_php_cookie';
unset($_COOKIE[$cookie_name]);
// empty value and expiration one hour before
$res = setcookie($cookie_name, '', time() - 3600);
```

Dec 2019 (Supple)

17. Write CSS style rules to implement the following in a web page: (3) -18
    a. to display the content of hyperlinks with yellow background colour and in italics
    b. to display the contents of unordered lists in bold and in Arial font
    c. to display a background image titled “birds.jpg” with no tiling.

a) a { background-color: yellow; font-style: italic; }
b) ul { font-weight: bold; font-family: Arial; }
c) body { background-image: url(“birds.jpg”); background-repeat: no-repeat; }

18. Discuss the various CSS style sheet levels with suitable examples. How are conflicts resolved when multiple style rules apply to a single web page element? (4) -19
    There are 3 levels of style sheets-

- Inline
  - is specified for a specific occurrence of a tag and apply only to that tag
  - appears in the tag itself
  - is fine-grain style, which defeats the purpose of style sheets - uniform style
- Document-level style sheets
  - apply to the whole document in which they appear
  - appear in the head of the document

- External style sheets
  - can be applied to any number of documents
  - are in separate files, potentially on any server on the Internet

- use a <link> tag to let the browser fetch and use an external style sheet file
  <link rel = "stylesheet" type = "text/css" href = "http://www.wherever.org/termpaper.css"></link>
- can be validated

When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence

- In a sense, the browser searches for a style property specification, starting with inline, until it finds one (or there isn't one)

19. Distinguish between descendant selectors and child selectors in CSS. If there is a style rule given by body p { color : red; font-family : courier; } and another style rule body > p { color : green; }, what styles will apply to the content of a paragraph in the web page body? (4) -27

Child Selector

The Child Selector is used to match all the elements which are children of a specified element. It gives the relation between two elements. The element > element selector selects those elements which are the children of a specific parent. The operand on the left side of > is the parent and the operand on the right is the children element.

Syntax:-

```
element > element {
   // CSS Property
}
```

Descendant Selector

Descendant selector is used to select all the elements which are child of the element (not a specific element). It selects the elements inside the elements i.e it combines two selectors such that elements matched by the second selector are selected if they have an ancestor element matching the first selector.

Syntax:-

```
element element {
   // CSS Property
}
```

For the given code, the output will be the contents inside the <p> tag with “green” coloured text and the font family will be “courier”. The child selector (body > p) overrides the descendant selector (body p). If a different font family had also been specified in the

child selector, then the font family of the output contents inside the <p> tag would also have been the one specified in the child selector.

20. What are class selectors in CSS? Suppose that it is required to display the text content of p tags in two different styles depending on the context.
    Style 1 – text should be displayed in blue colour, right aligned, underlined, with font style of italics and spacing between the words of text set to 2 cm.
    Style 2 – text should be displayed with a background colour of yellow, having a dashed, blue colour border, and with a padding of 1.5 cm
    Write the equivalent CSS style rules. (5) -34

The *.class* selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

You can also specify that only specific HTML elements should be affected by a class. To do this, start with the element name, then write the period (.) character, followed by the name of the class.

<style>

.class1{

Font-style: italic;

Word-spacing: 2cm

Text-align : right;

color:blue;

Text-decoration: underline;

}

.class2{

Background-color: yellow;

Border: dashed blue;

```
Padding: 2.5cm;

}

</style>

<p class=”class1”> Type 1</p>

<p class=”class2”>Type 2</p>
```

21. What is the function of cookies? Explain why session tracking is sometimes a better alternative to using cookies. (5) -38

- A session is the time span during which a browser interacts with a particular server.
- Cookies provide a general approach to storing information about sessions on the browser system itself.
- The server is given this information when the browser makes subsequent requests for resources from the server.
- A cookie is a small object of information that includes a **name** and a **textual value**.
- A cookie is created by some system on the server.
- The header part of an HTTP communication includes cookies.
- At the time it is created, a cookie is assigned a lifetime.
- When the cookie reaches its lifetime, the cookie is deleted from the browser’s host machine.

Why is session tracking better?

- In some cases, information about a session is **needed only during the session**.
- Also, the needed information about a client is nothing more than a unique identifier for the session—commonly used in shopping cart applications
- For these cases, a different process, named session tracking, can be used.
- Rather than using one or more cookies, **a single session array** can be used to store information about the previous requests of a client during a session.
- In particular, **session arrays often store a unique session ID for a session**.
- One significant way that session arrays differ from cookies is that they can be **stored on the server**, whereas cookies are stored on the client.

Session Tracking in PHP

- In PHP, a session ID is an **internal value that identifies the session**.
- Session IDs need not be known or handled in any way by PHP scripts.
- PHP is made aware that a script is interested in session tracking by calling the **session_ start function**, which takes no parameters.

- The first call to session_start in a session causes a session ID to be created and recorded. On subsequent calls to session_start in the same session, the function retrieves the **$_SESSION array**, which stores any session variables and their values that were registered in previously executed scripts in that session.
- Session key/value pairs are created or changed by assignments to the $_SESSION array.
- The session details can be destroyed with the unset operator.

22. How is XML different from HTML? What is meant by the term namespace in the context of XML? (5) -41

HTML vs XML

← prev next →

There are many differences between HTML (Hyper Text Markup Language) and XML (eXtensible Markup Language). The important differences are given below:

No.	HTML	XML
1)	HTML is used **to display data** and focuses on how data looks.	XML is a software and hardware independent tool used **to transport and store data**. It focuses on what data is.
2)	HTML is a **markup language** itself.	XML provides a **framework to define markup languages**.
3)	HTML is **not case sensitive**.	XML is **case sensitive**.
4)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.
5)	HTML **has its own predefined tags**.	You **can define tags according to your need**.
6)	In HTML, it is **not necessary to use a closing tag**.	XML **makes it mandatory to use a closing tag**.
7)	HTML is **static** because it is used to display data.	XML is **dynamic** because it is used to transport data.
8)	HTML **does not preserve whitespaces**.	XML **preserve whitespaces**.

- A Namespace is a set of unique names. Namespace is a mechanism by which element and attribute name can be assigned to a group. The Namespace is identified by URI(Uniform Resource Identifiers)
- .XML Namespaces provide a method to avoid element name conflicts.
- In XML, element names are defined by the developer.
- This often results in a conflict when trying to mix XML documents from different XML applications.
- The following XML carries table with rows and columns (May result in conflict):
  <table>
  <tr>
  <td>Apples</td>
  <td>Bananas</td>
  </tr>
  </table>

Namespace Declaration

A Namespace is declared using reserved attributes. Such an attribute name must either be xmlns or begin with xmlns: shown as below −

```
<element xmlns:name = "URL">
```

Syntax

- The Namespace starts with the keyword xmlns.
- The word name is the Namespace prefix.
- The URL is the Namespace identifier.
- Example (SOLUTION TO NAMECONFICTS)

```
<root>
<h:table xmlns:h="http://www.w3.org/TR/html4/">
<h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
</h:table>
<f:table xmlns:f="https://www.w3schools.com/furniture">
<f:name>African Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>
</root>
```

23. During the process of fetching a web page from a web server to a client browser, at what point does an embedded PHP script get executed? Can a client see the PHP script in the retrieved page? How does this differ from JavaScript script Execution? (3) -48

- The web server, on receiving a request, fetches the corresponding page from its disk.

- With the page now in memory, the web server notices that it is a file incorporating PHP scripting and passes the page to the PHP interpreter.
- The PHP interpreter executes the PHP code, including any SQL queries it makes etc.
- The PHP interpreter returns the results of the executed PHP code to the web server. The result of PHP execution is a pure HTML page without PHP code in it.
- The web server returns the page to the requesting client, which displays it.

No, a client cannot see the PHP script in the retrieved page.This is because PHP is a server side language.The server will have already taken care of the PHP, and what gets sent to the client is the resulting pure HTML.

24. What are the two modes that the PHP processor operates in? Explain. (2) -51

    The **PHP processor** has **two modes**, copy and interpret. The **processor** starts off in copy **mode**; text from the file, presumably HTML, is copied to the network connection and sent to the client browser.
    It takes a PHP document file as input and produces an HTML or XHTML document file.
    When the PHP processor finds markup code in the input file- it simply copies it to the output file.
    When the processor encounters PHP script in the input file- it interprets it and sends any output of the script to the output file.
    The output file is sent to the requesting browser

25. What is meant by a DTD? Create a DTD for a catalogue of cars, where each car element has the child elements make, model, year, colour, engine, doors. The engine element has the child elements no_of_cylinders and fuel_type. (5) -55

    - DTD stands for **Document Type Definition**.
    - DTD is a set of structural rules called **declarations**.
    - They specify a set of elements and attributes that can appear in a document, as well as how and where these elements and attributes may appear.
    - I.e., A DTD defines the structure and the legal elements and attributes of an XML document.
    - DTDs also provide entity definitions.
    - DTDs are used when the same tag set definition is used by a collection of documents.
    - An XML document with correct syntax is called "Well Formed".
    - An XML document validated against a DTD is both "Well Formed" and "Valid".
    - An application can use a DTD to verify that XML data is valid.

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

```
<!-- cars.dtd - a document type definition for
                the cars.xml document
  -->

<!ELEMENT car_catalog (car+)>
<!ELEMENT car (year, make, model, color, engine,
                    doors)>
<!ELEMENT make (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT engine (no_of_cylinders, fuel_type)>
<!ELEMENT doors (#PCDATA)>
<!ELEMENT no_of_cylinders (#PCDATA)>
<!ELEMENT fuel_type (#PCDATA)>

<!ENTITY c "Chevrolet">
<!ENTITY f "Ford">
<!ENTITY d "Dodge">
<!ENTITY h "Honda">
<!ENTITY n "Nisson">
<!ENTITY t "Toyota">
```

26. How is the type of a variable determined in PHP? Distinguish between bound and unbound variables. How can a variable be tested to check whether it is bound? (3) -58

The gettype() function is an inbuilt function in PHP which is used to get the type of a variable. It is used to check the type of existing variable.

Syntax:- `string gettype ( $var )`

This function accepts a single parameter *$var*. It is the name of variable which is needed to be checked for type of variable.

- A variable that is assigned a value before use is called bound variable
- An unassigned variable (*unbound variable*) has the value of NULL
- Unbound variables appearing on an expression have their value coerced
  - To 0 if the context specifies an integer
  - To the empty string if the context specifies a string

The isset() function checks whether a variable is set, which means that it has to be declared and is not NULL.This function returns true if the variable exists(bound variable) and is not NULL, otherwise it returns false(unbound variable).

Eg:-

```php
<?php

$var1 = 'test';

var_dump(isset($var1));

?>
```

Output will be true

27. What is the purpose of the explode() and implode() functions in PHP? Illustrate with suitable examples. (2) -1

The implode() function returns a string from elements of an array. It takes an array of strings and joins them together into one string using a delimiter (string to be used between the pieces) of your choice.

```php
<html>
<h3>Implode Function</h3>
<?php
$arr=array ('I','am','simple','boy!');
echo implode(" ",$arr);
?>
</body>
</html>
```

O/p ->

Implode Function

I am simple boy!

The explode() function in PHP allows us to break a string into smaller text with each break occurring at the same symbol. This symbol is known as the delimiter.

```php
<html>
<h3>Explode Function</h3>
<?php
```

```
$str="I am simple boy!";
print_r(explode(" ",$str));
?>
</body>
</html>
```

O/p ->

Explode Function

Array([0]=>I [1]=>am [2]=>simple [3]=>boy!)

28. What are the benefits of XML schema over DTDs? Discuss, with examples, the various user defined data types permissible for XML schemas. (5) -4

QUESTION

What are the benefits of XML schema over DTDs? Discuss with examples, the various user defined data types permissible for XML schemas.

ANSWER

The benefits of XML schemas are as follows:

(i) XML schema use basic XML syntax: XML schemas are created by using XML syntax whereas DTD's use separate syntax.

(ii) XML schema support namespace: XML schemas support namespace functionality, but DTD's don't support this functionality completely.

(iii) XML schema allow the validation of text elements based on datatypes: XML schemas specify the type of textual data that can be used within attributes & elements. This is done by the simple type declarations.

(iv) XML schema allow the creation of complex & reusable content models easily: In a DTD, content model can be reused only when the utilization of parameter entities is allowed. But, XML schemas provide a wide variety of mechanisms to reuse the content models & also model some complex programming concepts easily.

Data Types

XML schema data types can be generally categorized as "simple type" (including embedded simple type) & "complex type". The "embedded simple type" is already defined, but can be used to create a new type through restriction or extension.

Simple Type

A simple type is a type that only contains text data when expressed

according to XML 1.0. This type can be used with element declarations & attribute declarations. The embedded simple type is provided for in XML Schema Part 2. A restriction may be placed on an embedded simple type to create a new, unique simple type.

eg: <xs:element name="Department" type="xs:string"/>

Here, the section described together with "xs:string" is an embedded simple type according to XML Schema. In this example, we have established the definition that the data type for the element called "Department" is a text string.

Complex Type

A complex data type is a type that has a child element or attribute structure when expressed according to XML 1.0. An element declaration may be used with this type. There are no predefined complex type data types, so the user will always define their own.

```
eg: <xs:complexType name="EmployeeType">
      <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Name"/>
        <xs:element ref="Department"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Department" type="xs:string"/>
```

In this case the type name "EmployeeType" is designated by the name attribute of the complexType element. A model group is designated in the

child element.

New types are created by placing restrictions on or extending simple or complex types.

29. Write an embedded PHP script which displays the factorial of all numbers from 1 to 10 in a table in the web page. The factorial should be calculated and returned from a function. The table headings should be “Number” and “Factorial”. (5) -17

```
<!DOCTYPE html>
<html>
<body>
<table align="left" border="1" cellpadding="3" cellspacing="0">
<?php
function Factorial($number) {
$input = $number;
$fact=1;
for($i=$input; $i>=1;$i--) {
$fact = $fact * $i;
}
return $fact;
}
echo "<tr><th>Number</th><th>Factorial</th></tr>";
for($i=1; $i<=10; $i++) {
        $fact = Factorial($i);
        echo "<tr><th>".$i."</th><th>".$fact."</th></tr>";
}
?>
</table>
</body>
</html>
```

30. Write notes on entities in XML. State the format for declaring external entities (text and binary) in a DTD. (5) -18

Document is broken down into multiple logically related collections of information-entities.Document entity- it includes references to the names of entities that are stored elsewhere Every entity except the document entity must have a name. XML processor encounters the name of a nonbinary entity in a document, it replaces the name with the value it references.

Entity names can be any length.

- must begin with a letter, a dash, or a colon
- After the first character, a name can have letters, digits, periods, dashes,underscores, or colons.
- A reference to an entity is its name together with a prepended ampersand and an appended semicolon.

• Entities defined so that they can be referenced anywhere in the content of an XML document- general entities.

• Predefined entities are all general entities.

• Entities defined so that they can be referenced only in DTDs- parameter entities

• The form of an entity declaration is

<!ENTITY [%] entity_name "entity_value">

• the optional percent sign (%) is present in an entity declaration- specifies that the entity is a parameter entity .

• example of an entity

<!ENTITY mace "Mar Athanasius College of Engineering ">

• Any XML document that uses a DTD that includes this declaration can specify the complete name with just the reference &mace;.

• When an entity is longer than a few words, its text is defined outside the DTD- external text entity.

Entities can be primarily of four types −

- Built-in entities
- Character entities
- General entities
- Parameter entities

Format for external entities

The form of the declaration of an external text entity is

• <!ENTITY entity_name SYSTEM "file_location">

An external binary entity is defined as follows:

<!ENTITY entityName SYSTEM "fileName" NDATA JPEG>

31. What are associative arrays? How are associative arrays declared in PHP? Illustrate with an example. (2) -19

Associative arrays are arrays that use named keys that you assign to them. We can traverse associative arrays using loops. We can loop through the associative array in two ways. First by using for loop and secondly by using foreach.
There are two ways to create an associative array:
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
OR
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";

32. Briefly describe, with suitable examples, the various methods for iterating over the contents of a PHP array. (3) -27

- **foreach**

  PHP's foreach loop provides a convenient way to iterate over arrays. There are two forms: one uses both the key and value of each array entry while the other uses only the value.
  Eg:

```
$pets = ['Morie', 'Miki', 'Halo', 'lab' => 'Winnie'];

foreach ($pets as $key => $val) {
    echo "$key => $val \n";
}
```

- **list/each**

  Another common way to traverse arrays uses a while loop with the list language construct and the each function.The list/each construct can be used with both key and value, or just value.
  Eg:

```
$pets = ['Morie', 'Miki', 'Halo', 'lab' => 'Winnie'];

reset($pets); // reset array pointer
while ( list($key, $val) = each($pets) ) {
    echo "$key => $val \n";
}
```

- **for**

  A for loop can be used to iterate over numerically indexed arrays.
  Eg:

```
$ar = ['Rudi', 'Morie', 'Halo', 'Miki'];

for ($i=0, $len=count($ar); $i<$len; $i++) {
```

```
    echo "$ar[$i] \n";
}
```

PHP aldo provides several array iteration functions. They are array_walk, array_map & array_filter.

33. What is the purpose of XSLT style sheets? Suppose that an XML document contains a root element named students which has a child element student. Each student element has the child elements name, roll_no and branch. Suppose that there are 5 instances of the student element. Write an XSLT stylesheet to display the student data as an HTML document. (5) -34

eXtensible Stylesheet Language (XSL) is a family of recommendations for defining the presentation and transformations of XML documents. XSLT style sheets are used to transform XML documents into different forms or formats. XSLT transforms XML documents into XHTML documents, primarily for display.

ANSWER

XSLT, Extensible Stylesheet Language Transformations, provides the ability to transform XML data from one format to another automatically.

An XSLT stylesheet is used to define the transformation rules to be applied on the target XML document. XSLT stylesheet is written in XML format. XSLT processor takes the XSLT stylesheet & applies the transformation rules on the target XML document & then it generates a formatted document in the form of XML, HTML, or text format. This formatted document is then utilized by XSLT formatter to generate the actual output which is to be displayed to the end-user.

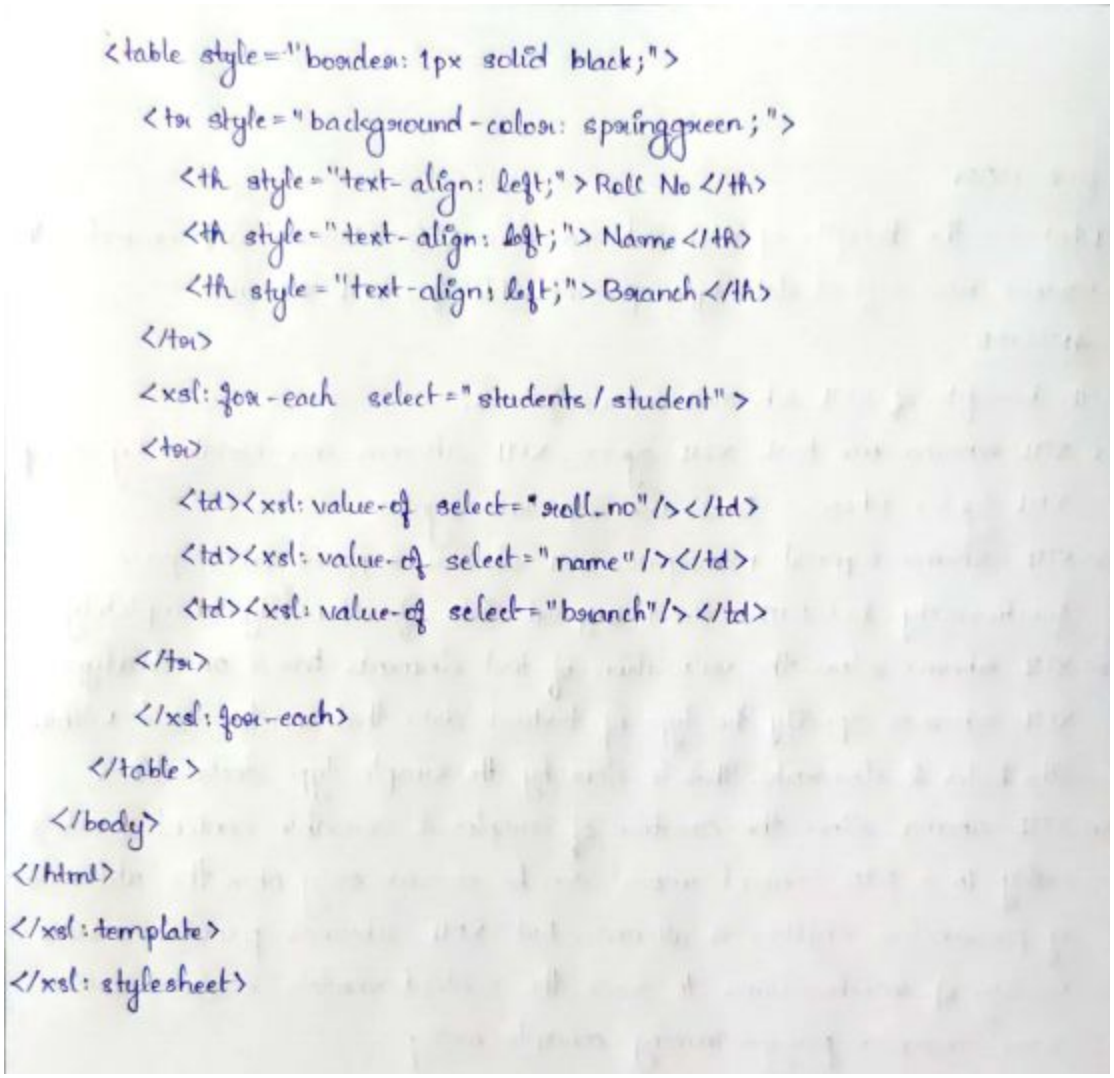
XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="student.xsl"?>
<students>
  <student>
    <roll_no>1</roll_no>
    <name>Abhijith G Nair</name>
    <branch>CS</branch>
  </student>
```

```
<student>
    <roll_no> 27 </roll_no>
    <name> Harikrishnan V </name>
    <branch> CS </branch>
</student>
<student>
    <roll_no> 38 </roll_no>
    <name> Megha K C </name>
    <branch> CS </branch>
</student>
<student>
    <roll_no> 41 </roll_no>
    <name> Pariyanga P Kini </name>
    <branch> CS </branch>
</student>
<student>
    <roll_no> 55 </roll_no>
    <name> Srividya Krishnakumar </name>
    <branch> CS </branch>
</student>
</students>
```

XSLT Stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h2> Students Information </h2>
```



```
<table style="border: 1px solid black;">
  <tr style="background-color: springgreen;">
    <th style="text-align: left;">Roll No</th>
    <th style="text-align: left;">Name</th>
    <th style="text-align: left;">Branch</th>
  </tr>
  <xsl:for-each select="students/student">
  <tr>
    <td><xsl:value-of select="roll_no"/></td>
    <td><xsl:value-of select="name"/></td>
    <td><xsl:value-of select="branch"/></td>
  </tr>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

34. What is the purpose of the implicit arrays $_POST and $_GET in PHP? Consider that a web page displays a form containing two text boxes (named num1 and num2), where the user enters numeric data. Write a PHP script which collects this form data, finds the sum, difference and the product of the two numbers and then displays the same with suitable messages. Assume that the script is to be embedded in the web page specified by the action attribute of the form and that the method used when the form is submitted is GET. (5) -38

- PHP can be used for Form Handling
- The PHP script is embedded in an XHTML document
- PHP can be configured so that form data values are directly available as implicit variables whose names match the names of the corresponding form elements
- PHP uses implicit arrays for form values $_POST and $_GET

- These arrays have keys that match the form element names and values that were input by the client
    - eg. If a form has a text box named street and the form method is POST the value of that element is available in the PHP script as $_POST[“street”]

```
<html>
<body>
 <form action=”welcome.php” method=”get”>
        Number1: <input type=”number” name=”num1”><br>
        Number2: <input type=”number” name=”num2”><br>
        <input type=”submit”>
 </form>
<?php
        $sum = $_GET[“num1”] + $_GET[“num2”] ;
        $diff = $_GET[“num1”] - $_GET[“num2”];
        $product = $_GET[“num1”] * $_GET[“num2”];
   echo "Sum is ". $sum . "<br>";
   echo "Difference is " . $diff . "<br>";
   echo "Product is " . $product;
?>
</body>
</html>
```

35. Can CSS be used to display an XML document? Illustrate with an example. (5) -41

Yes, CSS can be used to display the contents of the XML document.

- **Basic steps in defining a CSS stylesheet for XML :**

  For defining the style rules for the XML document, the following things should be done :-

  1. Define the style rules for the text elements such as font-size, color, font-weight, etc.
  2. Define each element either as a block, inline or list element, using the display property of CSS.
  3. Identify the titles and bold them.

- **Linking XML with CSS :**

  In order to display the XML file using CSS, link XML file with CSS. Below is the syntax for linking the XML file with CSS:

  <?xml-stylesheet type="text/css" href="name_of_css_file.css"?>

Example 1.

In this example, the XML file is created that contains the information about five books and displays the XML file using CSS.

- **XML file :**

  Creating Books.xml as :-

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="Rule.css"?>
<books>
    <heading>Welcome To GeeksforGeeks </heading>
    <book>
        <title>Title -: Web Programming</title>
        <author>Author -: Chrisbates</author>
        <publisher>Publisher -: Wiley</publisher>
        <edition>Edition -: 3</edition>
        <price> Price -: 300</price>
    </book>
    <book>
        <title>Title -: Internet world-wide-web</title>
        <author>Author -: Ditel</author>
        <publisher>Publisher -: Pearson</publisher>
        <edition>Edition -: 3</edition>
        <price>Price -: 400</price>
    </book>
    <book>
        <title>Title -: Computer Networks</title>
        <author>Author -: Foruouzan</author>
        <publisher>Publisher -: Mc Graw Hill</publisher>
        <edition>Edition -: 5</edition>
        <price>Price -: 700</price>
    </book>
    <book>
        <title>Title -: DBMS Concepts</title>
        <author>Author -: Navath</author>
        <publisher>Publisher -: Oxford</publisher>
        <edition>Edition -: 5</edition>
        <price>Price -: 600</price>
```

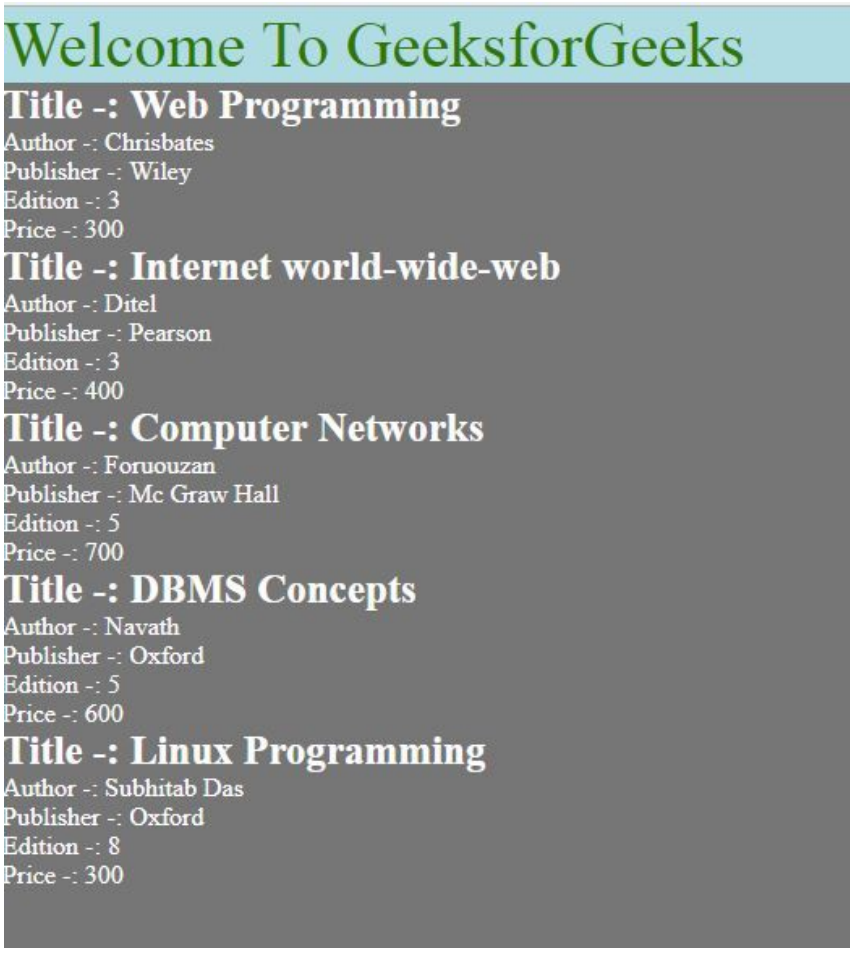
```
        </book>
        <book>
                <title>Title -: Linux Programming</title>
                <author>Author -: Subhitab Das</author>
                <publisher>Publisher -: Oxford</publisher>
                <edition>Edition -: 8</edition>
                <price>Price -: 300</price>
        </book>
</books>
```

In the above example, Books.xml is linked with Rule.css which contains the corresponding style sheet rules.

- **CSS FILE :**

  Creating Rule.css as:-

```
books {
        color: white;
        background-color : gray;
        width: 100%;
}
heading {
        color: green;
        font-size : 40px;
        background-color : powderblue;
}
heading, title, author, publisher, edition, price {
        display : block;
}
title {
        font-size : 25px;
        font-weight : bold;
}
```



May 2019

36. What is the speciality of pseudo selectors compared to other selector forms? (3) -48

Pseudo selectors often serve quite specific purposes.

Pseudo-classes are CSS classes used to define the state of an element. They target elements that can't be targeted with combinators or simple selectors like id or class. They are used to select elements based on their attributes, states, and relative position.They tend to act as if you had applied a class to some part of your document
A pseudo-class has a simple syntax. They are identified by a colon (`:`) placed just after a CSS selector.

It can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently

- Style an element when it gets focus

```
selector:pseudo-class {
  property: value;
}
```

When you hover over the link in the example, it will change color:

```
a.highlight:hover {
  color: #ff0000;
}
```

Pseudo-elements behave in a similar way, however they act as if you had added a whole new HTML element into the markup, rather than applying a class to existing elements. Pseudo-elements start with a double colon ::.

For example, if you wanted to select the first line of a paragraph you could wrap it in a element and use an element selector; however, that would fail if the number of words you had wrapped were longer or shorter than the parent element's width. As we tend not to know how many words will fit on a line — as that will change if the screen width or font-size changes — it is impossible to robustly do this by adding HTML.

The ::first-line pseudo-element selector will do this for you reliably — if the number of words increases and decreases it will still only select the first line.

```
article p::first-line {
    font-size: 120%;
    font-weight: bold;
}
```

37. What are the different ways of adjusting spacing in a text? Explain the various style attributes used for that. (3) -51

1. The letter-spacing property is used to specify the amount of space between letters. The amount indicated is in addition to the default spacing. The amount is specified in units. For example:
   Defines the spacing between the characters of a block of text.

   default letter-spacing: normal;

   The spacing between the characters is **normal**.

   letter-spacing: 2px;

   You can use **pixel** values.

   letter-spacing: 0.1em;

   You can use **em** values: this allows the spacing to remain *relative* to the font-size.

2. The word-spacing property is used to specify the amount of space between words. The amount indicated is in addition to the default spacing. The amount is specified in units. For example:
   It's a wide wide sentence!
3. The line-height property is used to specify the amount of vertical space between lines of text. The line-height can be specified in number of units, percentage, or with a multiplier.

4. The text-indent property is used to indent (or outdent) the first line of a block of text. The value can be specified in number of units or in percentage of the width of the containing block.

38. Write CSS code for setting the following styles (6) -55
    i) Table - A header row with background colour and bold font and other rows without that.
    ii) Unordered list – Set a smiley image as bullet.

i)

```
<style>
	th {
		background-color: #1a2b3c;
		font-weight: bold;
	}
</style>
```

ii)

```
<style>
	ul {
		list-style-image: src('smiley_image.png');
	}
</style>
```

39. What are the advantages of using a CSS framework? Explain the features of Bootstrap in relation to the above answer. (4) -58

Advantages of css framework

- Increased productivity—Reusing a structure is *more efficient* than re-coding every project from scratch.
- Standardized codebase—By default frameworks lead to a standard codebase as you're reusing the same code from project to project. This makes maintenance quicker and easier as you don't need to relearn how you developed each site.
- Makes it easier to work with a team—Similar to the above a standard codebase makes it easier for different people to work together on the same project.
- Provides a more organized approach to css—Abstracting design patterns and code blocks forces your code to become *more organized*.
- Fewer mistakes—Once you know a block of code or a larger pattern of code works you can safely add it to any project and know it won't contain *errors*.
- Cross browser reliability—Because they need to be the structure for any number of projects, frameworks tend to work for a variety of solutions including *multiple browsers*.

Bootstrap is a free front-end framework for faster and easier web development.

Features of bootstrap

- Easy to use - Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

- Responsive features - Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Simple Integration - Bootstrap can be simply integrated along with distinct other platforms and frameworks, on existing sites and new ones too and one more things you can also utilize particular elements of Bootstrap along with your current CSS.
- Pre-styled Components - Bootstrap approaches with pre-styled components for alerts, dropdowns, nav bars, etc.
- Customizable Bootstrap - The Bootstrap can be customized as per the designs of your project.
- Browser compatibility - Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)

40. External style sheets are usually preferred over other kinds of style sheets. Why? (4) -1

Cascading Style Sheets (CSS) are files with styling rules that govern how your website is presented on screen. CSS rules can be applied to your website's HTML files in various ways. You can use an **external stylesheet**, an **internal stylesheet**, or an **inline style**. Each method has advantages that suit particular uses.

An **external stylesheet** is a standalone .css file that is linked from a web page. The advantage of external stylesheets is that it can be created once and the rules applied to multiple web pages. Should you need to make widespread changes to your site design, you can make a single change in the stylesheet and it will be applied to all linked pages, saving time and effort.

41. Describe the schema of a document implemented in JSON and state how it is different from XML schema. (6) -4

Employees.json:

```
{
   "employees":[
      {
         "firstName":"John",
         "lastName":"Doe",
         "age":22,
         "is_manager":false
      },
      {
         "firstName":"Anna",
         "lastName":"Smith",
```

```
            "age":21,
            "is_manager":false
        },
        {
            "firstName":"Peter",
            "lastName":"Jones",
            "age":35,
            "is_manager":true
        }
    ]
}
```

This json document has an attribute employee whose value is an array of objects containing the details of employees with attributes firstName(string), lastName(string), age(number) and is_manager(boolean). The xml of the same doc is given below,

Employees.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<employees>
   <employee>
      <firstName>John</firstName>
      <lastName>Doe</lastName>
      <age>22</age>
      <is_manager>false</is_manager>
   </employee>
   <employee>
      <firstName>Anna</firstName>
      <lastName>Smith</lastName>
      <age>21</age>
      <is_manager>false</is_manager>
   </employee>
   <employee>
      <firstName>Peter</firstName>
      <lastName>Jones</lastName>
      <age>35</age>
      <is_manager>true</is_manager>
   </employee>
</employees>
```

Differences:

JSON	XML

JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON is supported by most browsers.	Cross-browser XML parsing can be tricky
JSON has no display capabilities.	XML offers the capability to display data because it is a markup language.
JSON supports only text and number data type.	XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data.
Retrieving value is easy	Retrieving value is difficult
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit
A fully automated way of deserializing/serializing JavaScript.	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object.	The object has to be express by conventions - mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding.
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.

It does not provide any support for namespaces.	It supports namespaces.
It is less secured.	It is more secure than JSON.

42. What are the primitives supported by PHP? (4) -17

PHP has a total of **8** primitive data types:

- **Integers** − are whole numbers, without a decimal point, like 4195.
- **Doubles** − are floating-point numbers, like 3.14159 or 49.1.
- **Booleans** − have only two possible values either true or false.
- **NULL** − is a special type that only has one value: NULL.
- **Strings** − are sequences of characters, like 'PHP supports string operations.'
- **Arrays** − are named and indexed collections of other values.
- **Objects** − are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources** − are special variables that hold references to resources external to PHP (such as database connections).

43. Differentiate between simple and complex data types in XML. (4) -18

There are two categories of user-defined schema data types: simple and complex.
A simple data type is a data type whose content is restricted to strings
A simple type cannot have attributes or include nested elements.
a large collection of predefined data types is included in the category of simple data types.
A complex type can have attributes and include other data types as child elements.
Simple type elements are defined in an XML schema with the element tag, which is from the XMLSchema namespace. Prefix xsd is normally used for names from this namespace
Eg: <xsd:element name = “engine” type = “xsd:string” />

Complex types are defined with the complexType tag
The elements that are the content of an element-only element must be contained in an ordered group, an unordered group, a choice, or a named group.
Eg:

```
<xsd:complexType name = "sports_car">
  <xsd:sequence>
    <xsd:element name = "make"  type = "xsd:string" />
    <xsd:element name = "model"  type = "xsd:string" />
    <xsd:element name = "engine"  type = "xsd:string" />
    <xsd:element name = "year"  type = "xsd:decimal" />
  </xsd:sequence>
</xsd:complexType>
```

44. Design a simple HTML form to input a string and to display whether it is palindrome or not when the form is submitted. Use a PHP script. (6) -19

```
<html>
<head>
<title>Palindrome Program</title>
</head>
<body>
<form method="post" action="program1.php">
<input type="text" name="number" value="" />
<input type="submit" name="submit" value="Submit" />
</form>
<?php
if(isset($_POST['number'])) {
$input = $_POST['number'];
$reverse = strrev($input);
if($reverse == $input) {
echo $input . 'is a palindrome';
}
else{
echo $input. 'is not a palindrome';
}
}
?>
</body>
</html>
```

45. List the various data types used in JSON. (5) -27

In JSON, values must be one of the following data types:

- a string
- a number

- an object (JSON object)
- an array
- a boolean
- Null

JSON values cannot be one of the following data types:

- a function
- a date
- undefined

JSON Strings

Strings in JSON must be written in double quotes.
Eg:
{ "name":"John" }

JSON Numbers

Numbers in JSON must be an integer or a floating point.
Eg:
{ "age":30 }

JSON Objects

Values in JSON can be objects.
Eg:
{
"employee":{ "name":"John", "age":30, "city":"New York" }
}

JSON Arrays

Values in JSON can be arrays.
Eg:
{
"employees":["John", "Anna", "Peter"]
}

JSON Booleans

Values in JSON can be true/false.
Eg:
{ "sale":true }

JSON null

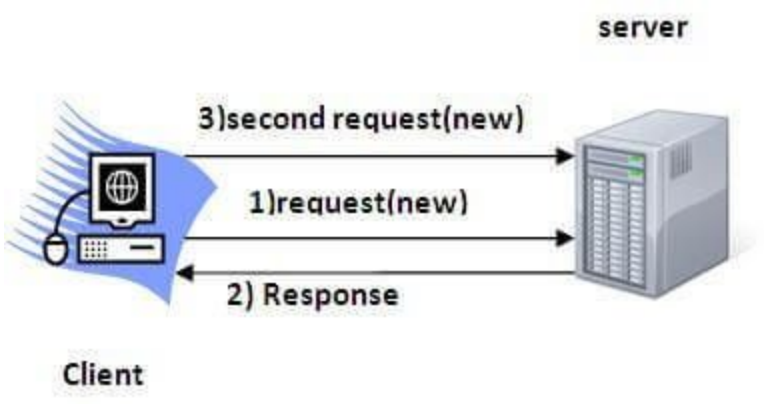
Values in JSON can be null.
Eg:
{ "middlename":null }

46. What are the contexts in which session tracking can be helpful? (5) -34

Session Tracking is a way to maintain state (data) of a user. It is also known as **session management** in servlets.

Http protocol is stateless so we need to maintain state using session tracking techniques. Each time a user requests the server, the server treats the request as the new request. So we need to maintain the state of a user to recognize a particular user.

HTTP is stateless, which means each request is considered as the new request. It is shown in the figure given below:



Why use Session Tracking?

To recognize the user It is used to recognize the particular user.

Session Tracking Techniques

There are four techniques used in Session tracking:

1. **Cookies**
2. **Hidden Form Field**
3. **URL Rewriting**
4. **HttpSession**

47. How does the use of cookies help in server-side scripting? Give examples for data stored using cookies. (5) -38

- A session is the time span during which a browser interacts with a particular server.
- Cookies provide a general approach to storing information about sessions on the browser system itself.
- The server is given this information when the browser makes subsequent requests for resources from the server.
- A cookie is a small object of information that includes a **name** and a **textual value**.
- A cookie is created by some system on the server.
- The header part of an HTTP communication includes cookies.
- At the time it is created, a cookie is assigned a lifetime.
- When the cookie reaches its lifetime, the cookie is deleted from the browser's host machine.

Eg1: One of the most common needs for information about a session is to implement shopping carts on websites. An e-commerce site can have any number of simultaneous online customers. At any time, any customer can add an item to or remove an item from his or her cart. Each user's shopping cart is identified by a session identifier, which could be implemented as a cookie. So, cookies can be used to identify each of the customers visiting the site at a given time.

Eg2: One common use of cookies is for a website to create profiles of visitors by remembering which parts of the site are perused by that visitor. Sometimes this is called personalization. Later sessions can use such profiles to target advertising to the client, in line with the client's past
interests. Also, if the server recognizes a request as being from a client who has made an earlier request from the same site, it is possible to present a customized interface to that client. These situations require that information about clients be accumulated and stored.

48. How can CSS be used to display a XML document? Illustrate with an example. (5) - 41

Refer q.35

49. Design an XML document containing details about four students. Student attributes can be added as per your choice. (5) -48

stud.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="stud.xsl"?>
<studInfo>
<stud>
<name>Abhilash</name>
<dept>IT</dept>
<rno>97709205001</rno>
</stud>
<stud>
<name>Akhila</name>
<dept>IT</dept>
<rno>97709205002</rno>
</stud>
<stud>
<name>Anaswara</name>
<dept>IT</dept>
<rno>97709205003</rno>
</stud>
<stud>
<name>Anu</name>
<dept>IT</dept>
<rno>97709205003</rno>
</stud>
```

```
</studInfo>
```

stud.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Student Details - Transformation</title>
</head>
<body>
<table border="1">
<caption>Student Details</caption>
<tr>
<th>Name</th><th>Department</th><th>Register No</th>
</tr>
<xsl:for-each select="/studInfo/stud">
<tr>
<td><xsl:value-of select="name"/></td>
<td><xsl:value-of select="dept"/></td>
<td><xsl:value-of select="rno"/></td>
</tr>
</xsl:for-each>
</table>
```

```
</body>

 </html>

</xsl:template>

</xsl:transform>
```

50. Write a PHP script to search for a particular string pattern in a text. (5) - 51

int preg_match (string pattern, string string [, array pattern_array], [, int $flags [, int $offset]]]);

Script -

```
<?php
   $line = "Vi is the greatest word processor ever created!";
   // perform a case-Insensitive search for the word "Vi"

   if (preg_match("/\bVi\b/i", $line, $match)) :
      print "Match found!";
      endif;
?>
```

This will produce the following result −

Match found!

OR

```
<?php
$search = 'foo';
$lines = file('file.txt');
// Store true when the text is found
$found = false;
foreach($lines as $line)
{
  if(strpos($line, $search) !== false)
  {
    $found = true;
    echo $line;
  }
}
// If the text was not found, show a message
```

```
if(!$found)
{
  echo 'No match found';
}
```

51. Illustrate with a sample PHP script, how functions are implemented in PHP. (5) -55

- A function is a block of code written in a program to perform some specific task.
- They take information as parameters, execute a block of statements or perform operations on these parameters and return the result.

- PHP provides us with two major types of functions:
    a. **Built-in functions**: PHP provides us with a huge collection of built-in library functions. These functions are already coded and stored in the form of functions. To use those we just need to call them as per our requirement.
    b. **User Defined Functions**: PHP also allows us to create our own customised functions called the user-defined functions.

Syntax:

```
function name([parameters])
{
      // function body: executable code
}
```

- **Creating a function:**
    1. Any name ending with an open and closed parenthesis is a function.
    2. A function name always begins with the keyword `function`.
    3. To call a function we just need to write its name followed by the parentheses.
    4. A function name cannot start with a number. It can start with an alphabet or underscore.
    5. A function name is **not case-sensitive**.
    6. The square bracket around the parameters means that they are optional.
    7. The function definition does not need to appear before it is called.
    8. The `return` statement is used to return a value from a function to the calling function and will be the last statement in a function.
- **Function Parameters or Arguments:**
    - The information or variable, within the function's parenthesis, are called **parameters**.
    - These are used to hold the values executable during runtime. A user is free to take in as many parameters as needed, separated with a comma(,) operator.

- While passing the values like during a function call, they are called **arguments**.
- Parameters in the call to a function: **actual parameters.**
- Parameters that are listed in the function definition: **formal parameters.**
- Parameters can be passed by **value** or by **reference** (the address of the variable in memory is passed, using an & sign before the variable name).

```
<?php

// function along with three parameters
function proGeek($num1, $num2, $num3)   // function definition
{
    $product = $num1 * $num2 * $num3;

    return $product; //returning the product
}

// storing the returned value
$retValue = proGeek(2, 3, 5); // function call
echo "The product is $retValue";

?>
```

- **Default Arguments:**
  - PHP allows us to set default argument values for function parameters.
  - If we do not pass any argument for a parameter with default value then PHP will use the default set value for this parameter in the function call.

e.g.:

```
<?php
function defGeek($str, $num=12) {
    echo "$str is $num years old \n";
}

// Calling the function
defGeek("Ram", 15);
// In this call, the default value 12  will be considered
defGeek("Adam");
?>
```

52. What is the role of DTD in an XML document? Illustrate with an example. (5) -58

DTD stands for Document Type Definition.An XML document validated against a DTD is both "Well Formed" and "Valid".With a DTD, independent groups of people can agree on a standard DTD for interchanging data.An application can use a DTD to verify that XML data is valid.A DTD defines the structure and the legal elements and attributes of an XML document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The DOCTYPE declaration above contains a reference to a DTD file. The content of the DTD file is shown and explained below.

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

The DTD above is interpreted like this:

- !DOCTYPE note - Defines that the root element of the document is note
- !ELEMENT note - Defines that the note element must contain the elements: "to, from, heading, body"
- !ELEMENT to - Defines the to element to be of type "#PCDATA"
- !ELEMENT from - Defines the from element to be of type "#PCDATA"
- !ELEMENT heading - Defines the heading element to be of type "#PCDATA"
- !ELEMENT body - Defines the body element to be of type "#PCDATA"

1, 4, 17, 18, 19, 27, 34, 38, 41, 48, 51, 55, 58