

Introduction to Information and Communication

Technologies for Development

Yukihiko Yamashita

Contents

- **Web system**
- World Wide Web (WWW)
- HTTP (HTML)
- CSS
- PHP

12 Web system

12.1 World Wide Web (WWW)

- Tim Berners-Lee
 - CERN (European Organization for Nuclear Research), beginning of 1990s
 - To display information on any platform (Unix, Mac, Dos, etc.)
 - The first homepage in Japan:
High Energy Accelerator Research Organization, 1992/9/30
- **HTTP** (Hypertext Transfer Protocol)
 - Hypertext : Text with hyper links
 - Server and client system.
 - Protocol to communicate Hypertext.

- **HTML** (Hyper Text Markup Language)
 - A language to describe hypertext (**tag language**)
Simplified from SGML (Standard Generalized Markup Language)
 - Link to another document, type of font, table, list can be described.
 - HTML 5.1 (The latest standard defined by w3C (World Wide Web consortium).)

12.2 Tag language

- A tag is written between < and >.
- A tag provides a meaning to text documents

Example of tag language:

```
<table border="1">  
  <tr><th>Item 1</th><th>Item 2</th></tr>  
  <tr><td>12.3</td><td>23.4</td></tr>  
</table>
```

12.3 CSS

- **Cascading Style Sheets**

- To separate data (Data Object Model) and expression.
- Size, type, and color of fonts, and align, span of table, and background color
- To show pages beautifully, they are specified in detail.

We will make HTML documents with CSS as a practice

12.4 Web browser

- **WorldWideWeb** : the first Web browser (1991)
 - Display a HTML document communicated by HTTP.
 - Renamed to Nexus later.
- Mosaic (1993) : It can display images.
NCSA (National Center for Supercomputing Applications)
- Branch Mosaic to Netscape Navigator (1994)
 - Javascript : A programming language worked on web browsers (1995).
- Mozilla Foundation (1996)
 - Netscape opened a source code.
 - **Firefox** (2002)
- **Internet explorer** (Microsoft)
- Chrome (Google), Safari (Apple), Opera
- Function of browser can be extended by Plug-ins
Because of HTML5 and security, plug-ins have not been used so much.

12.5 URL

- **URL** : Uniform Resource Locator
- All resources in a network are located by a common form.
- It can designate a web document.
- Example : `http://www.ide.titech.ac.jp:80/ja/index.php`
 - http : Scheme (protocol to communicate)
 - `www.ide.titech.ac.jp` (host/domain name)
the name of a machine in a network
 - 80 : Port number (UDP, TCP)
If the number is standard for the scheme, it can be abbreviated.
 - `/ja/index.php` : path and data name
- Examples of protocol: http, ftp, rcs, svn

12.6 Data of HTTP

- Data for HTTP are written in text.
- Data format for HTTP :
- **Header region**
 - We cannot see it without a plug-in.
 - It is not text between `<head>` and `</head>` in a html document.
 - Cookie
Information stored in a browser for each server.
- blank line
- **Data region**
 - Data in a html document
 - * title,
 - * character code,
 - * size and type of font,
 - * heading,

- * table,
 - * hyperlink,
 - * references to images,
 - * javascript,
 - * form.
- Binary data (such as an image) is transformed to text data.
MIME (Multipurpose Internet Mail Extensions)
 - Submitted data by POST
Values input by user are sent to the server.

12.7 Process of HTTP

- **HTTP Server**

- Apache
- IIS (Microsoft)

- **Process of data transaction by HTTP :**

1. Input URL by a browser
2. Search IP address of the server by DNS.
3. Send a request for data from a browser to the server. (GET)
If the browser has Cookie, it is sent to the server.
4. The server sends the data to the browser
5. The browser requests supplemental data in order to display the document.
(GET)
Image, css file, Javascript, etc. (The process may be iterated)
6. The server send the supplemental data by HTTP.

12.8 Web 2.0

- Tim O'Reilly claimed (2005)
- **Web** : One way from a sender at server side to a receiver.
- **Web 2.0** : **Participatory architecture**
Participants send information and participants receive it.

Reason of Web 2.0

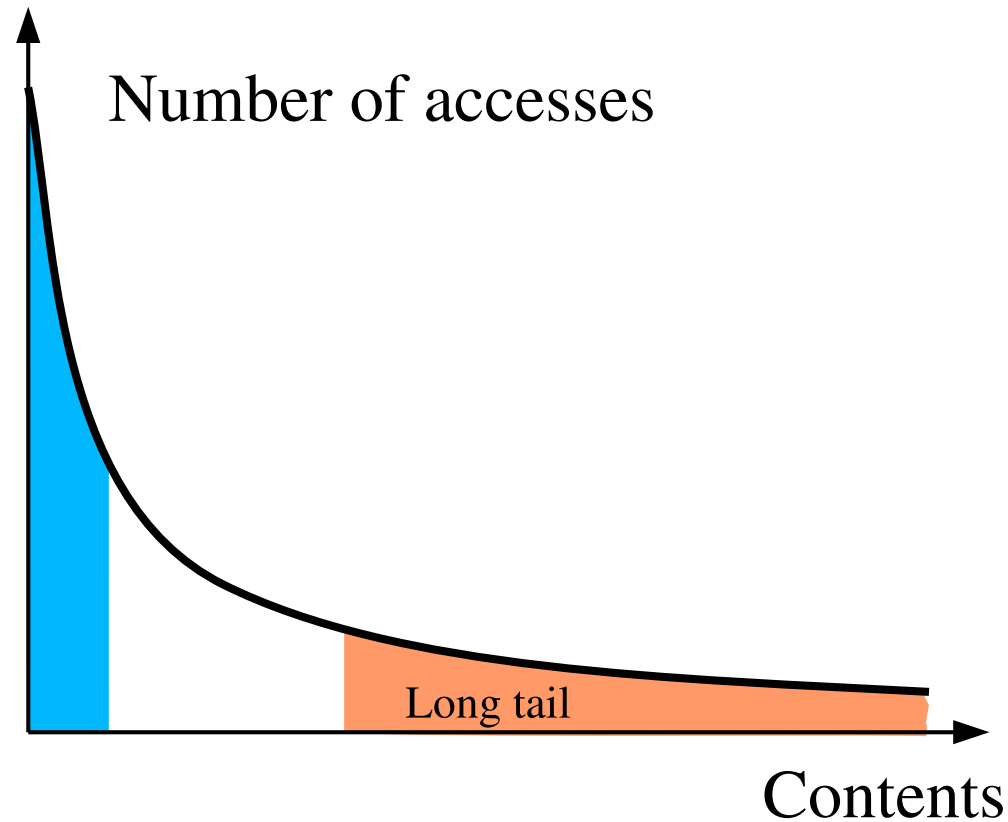
- Cost of communication is reduced :
Multimedia : Voice, image, and video (Youtube)
- Performance of computer is improved :
Information search for server side, Javascript for clients side
- Cost of storage is reduced :
Information classified to **long tail** can be stored and served.

Applications

- CMS (Contents Management System)
Almost CMSs are build by Aapche+MySQL+PHP

- e-Commerce, e-Education, e-Government, ...

12.9 Long tail



- Past : **20% of contents provide 80% of traffic.** (**Pareto principle**)
- Now : **More the 50%** are used for blog, auction, and SNS (Social Networking Service)

13 Technologies supporting Web 2.0

- **LAMP** (WAMP, MAMP)
 - Linux (Windows, Macintosh)
 - Apache : HTTP server (IIS)
 - * HTML
 - * CSS
 - MySQL : Data base (MariaDB, PostgreSQL, Oracle, DB2)
 - PHP : Programming language
 - PHP is used instead of cgi (common gateway interface) for Perl, Ruby, Python, Java, C
- **Javascript** (works on a web browser)
- **SSL** (Secure Sockets Layer) : Cryptography on transport layer
- **Ajax** (Asynchronous Javascript and XML)

13.1 MySQL

- **Relational database** (E. F. Codd, IBM, 1960)
- Data are managed by 2-dimensional table.
(Similar to Excel, Lotus 1-2-3, or Visicalc)
- At first, hierarchical database was considered to be promising so that relational database was not researched intensively.
- Relational algebra : Restriction, projection, and join
- Data structure are made by combining such simple tables.
- **Attempts to use a more complex data structure have not been succeeded.**
- Simple structure enables high speed searching and low cost maintenance.
- Furthermore, it is very **comprehensive**.

Examples of relational database table

grouptbl	
gid	name
1	East Asia
2	South East Asia
3	Africa

groupuser	
gid	pid
1	2
1	3
1	4
1	5
2	1
2	2
2	8
3	3
3	7
3	9

usertbl	
pid	name
1	Hirobumi ITO
2	Kiyotaka KURODA
3	Saneto SANJO
4	Aritomo YAMAGATA
5	Saneyoshi MATSUKATA
6	Shigenobu OKUMA
7	Kinmochi SAIONJI
8	Taro KATSURA
9	Gonbei YAMAMOTO

13.2 SQL

- **SQL** : Structured Query Language (not authorized)
- Data management language.
 - Search (select)
 - Update data (update)
 - Add rows (insert)
 - Delete rows (delete)
 - Manage of database (create, drop)
- Example : Search names of members in group 'East Asia'
select usertbl.name from grouptbl,groupuser,usertbl
where grouptbl.gid=groutuser.gid
and groutuser.pid=usertbl.PID
and grouptbl.name='East asia'
- Computer automatically optimizes the way of search in order to make search time shortest.
- Key, hash table, sort, etc.

13.3 PHP

- **Hypertext Preprocessor**

- An object oriented programming language processed on server.

Process of PHP

1. Demand is sent from a browser to a server. (GET, POST)

2. If demanded data is a HTML file, it is sent to the browser.

If it is a PHP file, the PHP program is executed and the output of the PHP program (print sentence, etc) is sent to the browser.

Furthermore,

- If data from a user (browser) to server is sent, it is passed to the PHP program.

Data is stored in `$_GET`, `$_POST` variables.

- If the PHP program has MySQL commands, database is searched/operated.

13.4 Javascript

- Netscape developed Javascript in 1995.
- It works as an interpreter on a browser
- cf. PHP, Java, Perl, etc. work on a server.
They send a HTML document to a browser.
- Javascript can make an **active page**.
- It is a scripting language.
PHP, Perl, Ruby, and Python are examples of scripting languages.
- Object oriented programming language
- There is no relation with programming language Java.
- It can refer/change/add HTML data and style.
- HTML data is managed by DOM (Document Object Model)
- It cannot access file in the computer for security.

13.5 Ajax

- **Asynchronous JavaScript and XML**
- XMLHttpRequest is implemented on Internet Explorer 5 in 1999.
- It enabled to asynchronously obtain data from a server and rewrite the display by Javascript.
- The technology has become common by Google Map (2005)
- **XML (eXtensible Markup Language)**
 - Structured text with tags similarly to HTML.
 - The grammar is strictly defined by DTD (Document Type Definition).
 - xhtml, docx, and xlsx are described by XML.
(Sometimes they are compressed by the zip coder).
- Not only with XML but also without XML, what uses asynchronous communication is used is called Ajax.
- JQuery : library for Ajax.

14 Practice

14.1 About practice

- Server : `www123.ide.titech.ac.jp`
(The server can be connected only from IP addresses of Tokyo Tech.)
- We assume the user name is `18M12345`.
- For login command : `ssh 18M12345@www123.ide.titech.ac.jp`
- URL for user web :
`http://www123.ide.titech.ac.jp/~18M12345`
- Copy files in `public_html` to the folder `public_html` of the server.
At home folder type the command:

```
rsync -av public_html/ 18M01234@www123.ide.titech.ac.jp:public_html/
```
- Download Visual Studio Code Editor (VS Code) from
`https://code.visualstudio.com/download`
 - User Finder to extract it from zip file.
 - Double click it.
 - Fix it on the task bar.

To do first.

- Login a computer (Mac) of GSIC.
- Open a terminal.
- Make a folder `public_html`
(HTML, CSS and PHP files have to be stored in the folder.)
- Download VS Code. (Install seems to be impossible.)

To do usually.

- Practice for HTML, CSS and PHP.
 - Make files in the folder `public_html` by VS Code (Select the folder).
 - Copy files to the server by `rsync` command
 - Watch the result by using web browser.
- Practice for SQL.
 - Open a terminal.
 - Login to the server.
 - Type commands in the terminal.

14.2 HTML

14.2.1 Basis of HTML

- The tag command is valid between `<(tag)>` and `</(tag)>`, where `(tag)` is name of tag.
- A tag command is denoted by `<(tag)>` below.
- System: `<head>`, `<body>`, `<title>`, `<meta>`,
- Headings: `<h1>`, `<h2>`, `<h3>`, ...
- List: `` or `` (with number)
- Item of list: ``

14.2.2 Table

- Table: `<table>`
- A row in Table: `<tr>`
- A data box in a row: `<td>`

14.2.3 Attribute

- Type of font: ``
 - Attribute for color: `color="#RRGGBB"` (`color="#FF0000"`)
 - Attribute for size: `font="(Number of size)"` (`size="4"`)
- In tags
 - Attribute for background color: `bgcolor="#RRGGBB"`
(ex. `<table bgcolor="#00FF00">`)
 - Attribute for align: `align="(align)"`
(ex. `<td align="center">`)

14.2.4 Practice of HTML 1, 2, and 3

List of pracH1.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Practice of HTML 1 (title)</title>
</head>
<body>
<h1>Practice of HTML 1</h1>
```

This is the first practice of HTML.

```
<ul>
<li>Hello world!</li>
<li>HTML is simple.</li>
<li>However, it has many commands.</li>
</ul>
</body>
</html>
```

List of pracH2.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Practice 2 (title)</title>
</head>
<body>
<h1>Practice 2</h1>

<b>This is the first practice of HTML</b>
<br></br> <!-- br is new line -->
<br/>      <!-- <br/> is the same as <br></br> -->
<table>
<tr><td>Day in a week</td><td>1, 2</td><td>3, 4</td></tr>
<tr>
<td>Monday</td><td>Introduction to Environment</td><td>Practice of Environme
</tr><tr>
<td>Tuesday</td><td>Introduction to Engineering</td><td>Introduction to Scie
</tr>
```



```
</table>  
</body>  
</html>
```

List of pracH3.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Practice of HTML 3 (title)</title>
</head>
<body bgcolor="#d0ffd0">
<h1>Practice of HTML 3</h1>

<h2>List</h2>

<ul>
<li>Hello world!</li>
<li><font color="a00080">HTML is simple.</font></li>
<li>However, it has many commands.</li>
</ul>

<h2>Table</h2>
```

```
<table border="2">
<tr><td>Day in a week</td><td>1, 2</td><td>3, 4</td></tr>
<tr>
<td>Monday</td><td bgcolor="#d0d0ff">Introduction to Environment</td><td>Pra
</tr><tr>
<td>Tuesday</td><td>Introduction to Engineering</td><td><font color="#ff0000
</tr>
</table>
<br>
<div align="center">
<font color="#0000ff" size="7">This is a pen.</font>
</div>

</body>
</html>
```

14.3 CSS

- **Cascading Style Sheets**

- To separate data (Data Object Model) and expression.
- Size, type, and color of fonts, and align, span of table, and background color
- To show pages beautifully, they are specified in detail.
- selector : basic selector (html tags, eg. `table`, `td`), ID (`#idName`),
class (`.className`)
- ID can specify a html tag.
- Class can specify html tags.
- attribute : to be set values (`font-size`, `background-color`, etc.)
- Format:

```
selector {attribute: value}
```

```
selector {attribute1: value2; attribute1: value2}
```

```
selector1,selector2 {attribute: value}
```

```
selector1 selector2 {attribute: value}
```

```
selector1>selector2 {attribute: value}
```

- Example:

```
tr {background-color: #ff0000}
table {background-color: #ff0000; border: 1}
td,th {font-size: 1.5em}
.className td {font-family: arial}
```

- `<div>` (division), `` tags are usually used only to specify style, ID, and class in a html document.

```
<div class="tableFormat1">
<table id="table1">

</table>
</div>
```

- Pseudo class in selector. (`basic-selector:pseudo-class`)
Example: `a:hover {color: #ff0000;}`
(‘hover’ means “mouse is on it”.)

- Privilege is (Cascading)
 - Describer > User > Browser
 - In line > Embedded style sheet > External
 - ID, attribute > Class > Type (html tag)

Use of CSS

- In line
``
- Embedded style sheet: In the header of a html document;
`<style>`
`td {color: #ff9900; background-color: #ff0000;}`
`</style>`
- External: In the header of a html document;
`<link rel="stylesheet" href="(file name).css" type="text/css">`

14.3.1 Practice of HTML and CSS

List of pracC1.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="pracC1.css" type="text/css">
<title>Practice of CSS 1 (title)</title>
</head>
<body>
<h1>Practice of CSS 1</h1>

<h2>List</h2>

<ul>
<li>Hello world!</li>
<li id="listnonaka">HTML is simple.</li>
<li>However, it has many commands</li>
</ul>
```

```
<h2>Table</h2>
```

```
<table>
```

```
<tr><td>Day in a week</td><td>1, 2</td><td>3, 4</td></tr>
```

```
<tr>
```

```
<td>Monday</td><td id="tablenonaka">Introduction to Environment</td><td>Prac
```

```
</tr><tr>
```

```
<td>Tuesday</td><td>Introduction to Engineering</td><td>Introduction to Scie
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<div class="hahahaha">
```

```
This is a pen.
```

```
</div>
```

```
</body>
```

```
</html>
```


List of pracC1.css

```
h1 {  
    background-color: #ff8080;  
    color: green;  
}
```

```
h2 {  
    background-color: #8080ff;  
    color: red;  
}
```

```
h2:hover {  
    background-color: #000000;  
    color: #0000ff;  
}
```

```
table {  
    border: solid 3px red;  
}
```

```
td {  
    color: green;  
    border: dashed 2px #204040;  
}
```

```
/* id selector */  
#listnonaka {  
    color: #a00080;  
}
```

```
#tablenonaka {  
    color: red;  
    background-color: #a0a0a0;  
}
```

```
/* class selector */  
.hahahaha {  
    text-align: center;  
    color: #0000ff;  
    font-size: xx-large;
```

}

14.4 Practice of SQL

Basis

- First, to login to server use
`ssh www123.ide.titech.ac.jp` and password.
- Login to database:
`mysql -u 00M00000 -p`
`(password)`
- Login :
`quit`
- Create database (Its name is 00M00000) :
`create database 00M00000;`
- Use a database :
`use 00M00000;`
- Delete a database :
`drop database 00M00000;`

Table

- Make a table :

```
create table testTbl (type char(1), sname text, fname text);
```

- Input data :

```
insert into testTbl values ('1', 'Einstein', 'Albert');  
insert into testTbl values ('1', 'Newton', 'Isaac');  
insert into testTbl values ('2', 'Curie', 'Maria');  
insert into testTbl values ('3', 'Heisenberg', 'Werner');
```

- Search data :

```
select * from testTbl;  
select type,sname from testTbl;  
select * from testTbl where type='1';
```

- Update data :

```
select * from testTbl;  
update testTbl set type='5' where sname='Curie';  
select * from testTbl;
```

- Delete data :

```
select * from testTbl;  
delete from testTbl where type='1';  
select * from testTbl;
```

- Delete a table :

```
drop table testTbl;
```

Tables

- Make tables :

```
create table groupTbl (gid char(4), name varchar(64));
create table groupUser (gid char(4), uid varchar(4));
create table userTbl (uid char(4), name varchar(64));
```

- Insert data to userTbl :

```
insert into userTbl values ('0001', 'Hirobumi ITO');
insert into userTbl values ('0002', 'Kiyotaka KURODA');
insert into userTbl values ('0003', 'Saneto SANJO');
insert into userTbl values ('0004', 'Aritomo YAMAGATA');
insert into userTbl values ('0005', 'Saneyoshi MATSUKATA');
insert into userTbl values ('0006', 'Shigenobu OKUMA');
insert into userTbl values ('0007', 'Kinmochi SAIONJI');
insert into userTbl values ('0008', 'Taro KATSURA');
insert into userTbl values ('0009', 'Gonbei YAMAMOTO');
```

- Insert data to groupTbl :

```
insert into groupTbl values ('0001', 'East Asia');
insert into groupTbl values ('0002', 'South East Asia');
insert into groupTbl values ('0003', 'Africa');
```

- Insert data to groupUser :

```
insert into groupUser values ('0001', '0002');
insert into groupUser values ('0001', '0003');
insert into groupUser values ('0001', '0004');
insert into groupUser values ('0001', '0005');
insert into groupUser values ('0002', '0001');
insert into groupUser values ('0002', '0002');
insert into groupUser values ('0002', '0008');
insert into groupUser values ('0003', '0003');
insert into groupUser values ('0003', '0007');
insert into groupUser values ('0003', '0009');
```

- Search with join :

```
select * from groupTbl,groupUser,userTbl
      where groupTbl.gid=groupUser.gid and userTbl.uid=groupUser.uid;
```

- Search data with join and limited columns :

```
select userTbl.name,groupTbl.name from groupTbl,groupUser,userTbl
      where groupTbl.gid=groupUser.gid and userTbl.uid=groupUser.uid;
```

- Search without join :

```
select userTbl.name,groupTbl.name from groupTbl,groupUser,userTbl;
```

15 PHP

15.1 Grammar

15.1.1 Basis of the grammar

- PHP program can be described in HTML document.
(Its suffix of file name is `.php`.)
- It starts by `<?php` and ends by `?>`.
- A line is ended by `;`.
- Example:

```
<?php
$a = 3;
$b = 3;
$c = $a + $b;
print $a;
?>
```


15.1.2 Variable

- Start with \$.
- Alphabet, numerals, and _ can be used.
- Numeral letter is not allowed after \$.
- The type of variable does not have to be defined.
- Example: \$a, \$abcd, \$number2ac

15.1.3 Array

- A type of variable.
- Example:

```
$arr1 = array();  
$arr1[0] = 123;  
$arr1[1] = 32.1;  
$arr1[3] = "abc";  
$arr1[4] = $arr1[0] + $arr1[1];  
print $arr1[4];
```

- Associative array. Its index is not integer but string.
- Example:

```
$arr1 = array();  
$arr1["this"] = 123;  
$arr1["number"] = 32.1;  
$arr1["alpha"] = "abc";  
$arr1["wa"] = $arr1["this"] + $arr1["number"];  
print $arr1["wa"];
```

- The array variable `$_SESSION` is used to store values for the session.
- The array variable `$_POST` and `$_GET` are used to obtain data that user inputs in a form.

15.1.4 Control

- Similar to C language.
- `if`, `else`, and `else if` can be used.
- `switch` and `case` can be used.
- `while` and `for` can be used.
- `foreach` can be used for the associative array.

15.1.5 String

- Single/Double quotation is used for the literal of string.
"This is a pen" or 'This is a pen'
- Concatenated by period.
"This is ". "a pen". " ." is equal to "This is a pen." .
"This is " . 4 . " ." is equal to "This is 4." . (A number is changed to a string and concatenate.)
- `print((string))` : Print string as a content of the html document.

15.2 Practice of PHP

List of pracPVar1.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP Variable 1 (title)</title>
</head>
<body>
<h1>Practice of PHP Variable 1</h1>
```

This is the first practice of PHP.

```
<ul>
<li>Hello world!</li>
<?php
    $list = "<li>PHP is simple.</li>";
    print $list;
    $a = 1;
    $b = 12.3;
```

```
$c = $a + $b;
```

```
?>
```

```
<li>However, it has many commands</li>
```

```
</ul>
```

```
$c =
```

```
<?php
```

```
    print $c;
```

```
?>
```

```
</body>
```

```
</html>
```

List of pracPVar2.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP Variable 2(title)</title>
</head>
<body>
<h1>Practice of PHP Variable 2</h1>

<?php

$i = 4;
$j = 10;
$k = $j / $i;

print("$k = " . $k . "<br> \n");
print("$k = " . (int)$k . "<br> \n");

print('sin($k) = ' . sin($k) . "<br> \n");
```

```
print("sin($k) = " . sin($k) . "<br> \n");
```

```
print('aaa \n');  
print(" xxx\n");
```

```
$l = $j % $i;  
print('$l = ' . $l . "<br> \n");
```

```
$k = 'abcd';  
$l = "efgh\n";
```

```
print($k.$l);
```

```
?>
```

```
</body>
```

```
</html>
```

List of pracPArray1.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP Array 1 (title)</title>
</head>
<body>
<h1>Practice of PHP Array 1</h1>

<?php

$array1 = array();
$array1[0] = 0;
$array1[1] = 11;
$array1[2] = 22;
$array1[3] = 33;
$array1[4] = 44;

for ($i = 0 ; $i < 5 ; ++$i) {
```



```
    print('$array1['.$i.'] = '. $array1[$i]. " <br>\n");
}

$array2 = array();
$array2["class"]      = "gakkyu";
$array2["student"]    = "Seito";
$array2["information"] = "Jouhou";

foreach ($array2 as $english => $japanese) {
    print('$array2['.$english.'] = '. $japanese. " <br>\n");
}

?>
</body>
</html>
```

Make your original PHP program using the example in Array.
(`pracPArray2.php`.)

List of pracPIf.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP 2 (title)</title>
</head>
<body>
<h1>Practice of PHP If</h1>

<?php

$x = 3;
$y = 5;

if($x > 2) {
    if($y > 3) {
        printf("x is larger than 2 and y is larger than 3\n");
    } else {
        printf("x is larger than 2 and y is not larger than 3\n");
    }
}
```

```
    }  
} else {  
    printf("x is not larger than 2\n");  
}
```

```
?>
```

```
</body>
```

```
</html>
```

List of pracPFor.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP 2 (title)</title>
</head>
<body>
<h1>Practice of PHP For</h1>

<?php

$sum = 0;
for ($i = 1 ; $i <= 100 ; ++$i) {
    $sum = $sum + $i;
}

print("The sum from 1 to 100 is " . $sum);

?>
```

```
</body>  
</html>
```

List of pracPPrime.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP Prime Numbers (title)</title>
</head>
<body>
<h1>Practice of PHP Prime Numbers</h1>

<?php

$nPrime = 0;
for ($primeCand = 2 ; $primeCand <= 1000 ; ++$primeCand) {
    $nDivided = 0;
    for ($divisor = 1 ; $divisor <= $primeCand ; ++$divisor) {
        if ( $primeCand % $divisor == 0)  ++$nDivided;
    }
    if ($nDivided ==2) {
        ++$nPrime;
    }
}
```

```
    print($nPrime . "-th prime number is ". $primeCand . " <br>\n");  
  }  
}
```

```
?>
```

```
</body>
```

```
</html>
```


List of pracPForm.php.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice of PHP Form (title)</title>
</head>
<body>
<h1>Practice of PHP Form</h1>
```

Addtion

```
<form name="form1" method="post" action="pracPForm.php">
1st number: <input type="text" name="num1"></input><br>
2nd number: <input type="text" name="num2"></input><br>
<input type="submit" name="execAdd" value="Add"></input><br>
</form>
```

```
<?php
if (isset($_POST["execAdd"])) {
```

```
$n1 = (int) $_POST["num1"];  
$n2 = (int) $_POST["num2"];  
printf("%d + %d = %d <br>\n", $n1, $n2, $n1 + $n2);  
}  
?>  
</body>  
</html>
```

Make a PHP program with form for prime number decomposition.
(`pracPPrimeDeconv.php`.)

- Use database from PHP program.
- `$mysqli = new mysqli()` : Connect to database.
- `$result = $mysqli->query()` : Execute a SQL sentence.
- `$row = $result->fetch_array ()` : Fetch a row from the result.

List of `pracSQL.php`.

```
<?php
$error = False;
$htmlStr = '';
$mysqli = new mysqli('localhost', '00M00000', '(password)', '00M00000');
/* 接続状況をチェックします */
if ($mysqli->connect_errno) {
    $htmlStr .= "<font color=\"red\">".$mysqli->connect_error."</font>";
    $error = True;
} else {
    $mysqli->set_charset ( "utf8" );
}

if (! $error) {
    $query = "select userTbl.name, groupTbl.name from groupTbl,groupUser,userTbl where groupTbl.gid=groupUser.gid and userTbl.uid=groupUser.uid;";
    if ($result = $mysqli->query ( $query )) {
        $nRow = $result->num_rows;
        $htmlStr .= "The number of combination of member and group is " . $nRow . "<br>\n";
        $htmlStr .= "<table border=\"1\">\n";
        $htmlStr .= "<tr><td>Name</td><td>Group</td></tr>\n";
        while ( $row = $result->fetch_array () ) {
            $htmlStr .= "<tr><td>" . $row [0] . "</td><td>" . $row[1] . "</td></tr>\n";
        }
        $htmlStr .= "</table>\n";
    }
}
```

```
?>

<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Practice SQL (title)</title>
</head>
<body>
<h1>Practice SQL</h1>
<?php
echo $htmlStr
?>
</body>
</html>
```

16 Conclusion

- Web system is one of the most important information providing systems.
- Web system can be used in developing countries and it will help to develop such countries in many regions.
- In this lecture, **basic technologies** for Web system are presented. Internet, HTTP, HTML, Web browser, URL, Database, SQL, PHP, CSS, Javascript, Ajax
- Web system is also improving for convenient and dependable usage.
- Very basic practices of Web system.

From the next class, **Prof. Aoki** will provide lectures at **S611 lecture room**.