Лекція 2

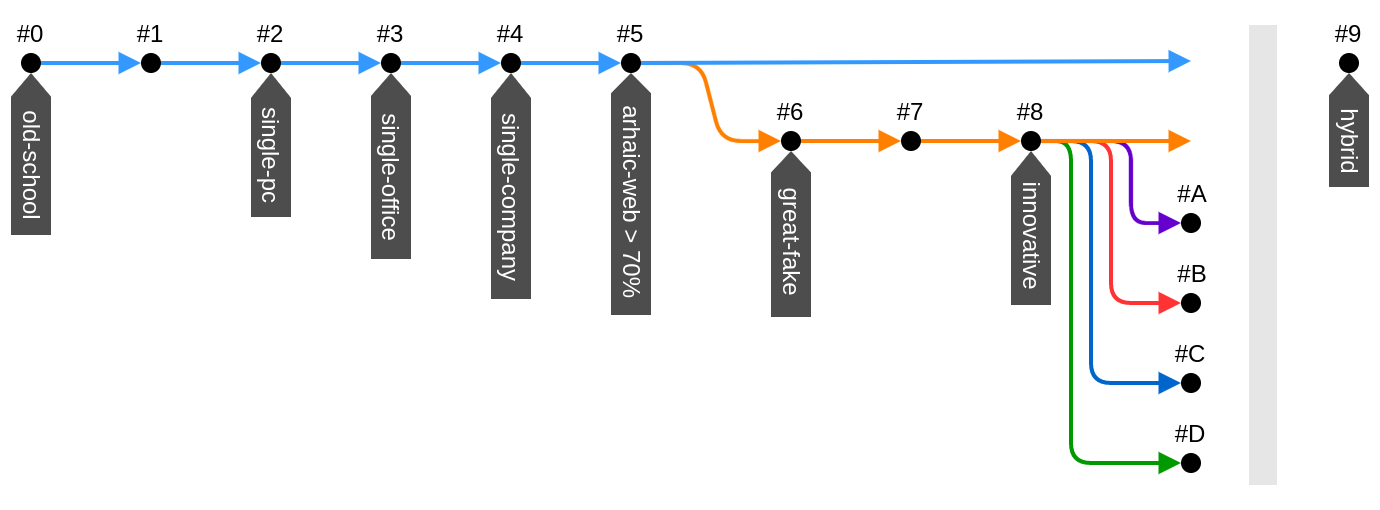
**Основи web-програмування**

План.

1. Еволюція додатків
2. Базові технології web-розробки.
3. Технологія CGI.
4. Історія розвитку PHP.
5. Ствоення простої форми.

**Еволюція додатків**

ІС - це системи, що забезпечують обробку, передачу та зберігання даних. Це далеко не всі програмування, але зараз ІС найчастіше асоціюються з веб і мобільними додатками, хоча і не збігаються з ними повністю, знак рівності між UI і ІС не можна ставити тим більше. Розгалуження в технологіях траплялися багато разів в історії, але за періодом невизначеності завжди слід етап сталого розвитку, коли несуттєві гілки йдуть в тінь, а актуальні особливості об'єднуються в найжиттєздатніший гібрид.



Простежимо, як ІС розвивалися з самого їх появи.

***IPC*** *- Взаємодія між процесами (англ. Inter-Process Communication) — набір засобів обміну повідомленнями між процесами.* *Для взаємодії процесів, що виконуються на одному комп'ютері (під управлінням однієї операційної системи) механізми взаємодії забезпечуються ядром операційної системи, в якій виконуються процеси.*

***RPC*** *(Remote Procedure Call), віддалений виклик процедур — технологія, що забезпечує взаємодію між процесами подібно до виклику функцій, дані в один бік передаються як аргументи функцій (віддалених процедур), в іншому — як результати виконання функцій (віддалених процедур).*

***SSE*** *(від англ. Server-Sent Events - «події, що посилаються сервером») являє собою технологію відправки повідомлень від сервера до веб-браузеру у вигляді DOM-подій. Технологія Server-Sent Events зараз стандартизується як частина HTML5 [1] організацією W3C*

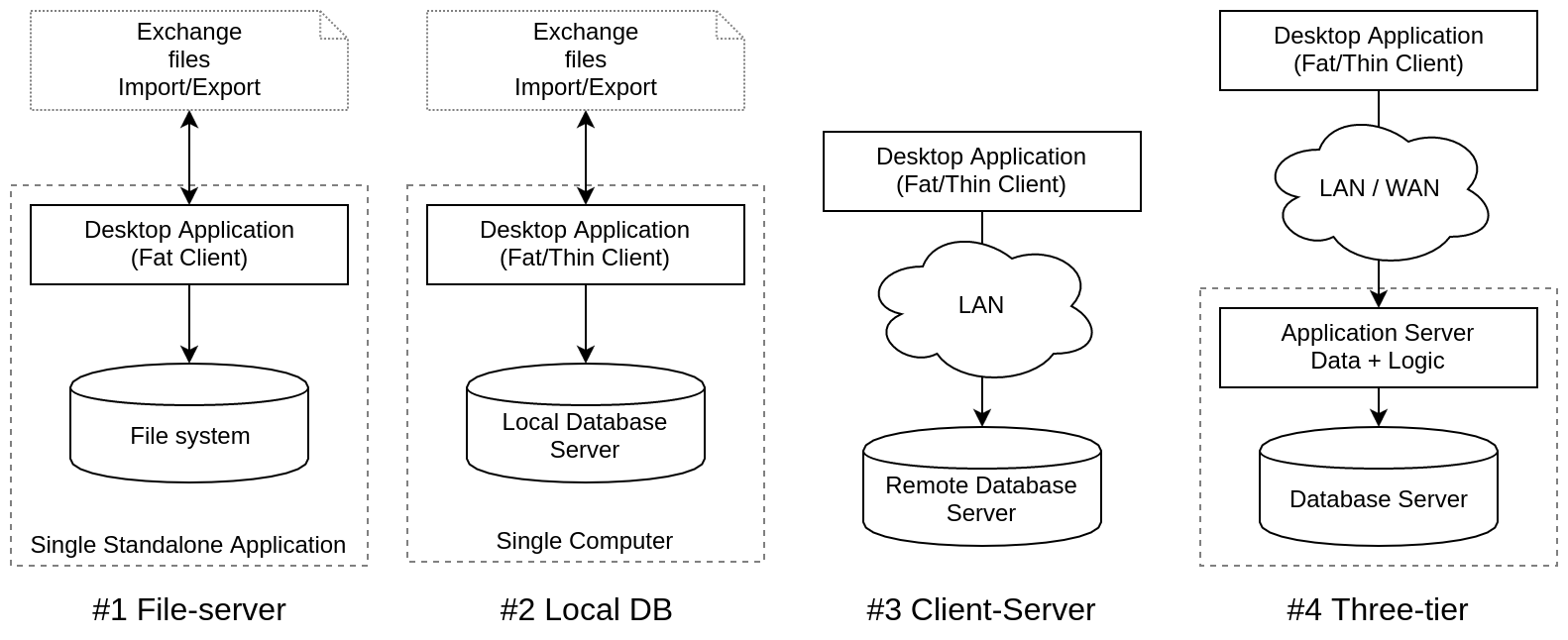
***Comet*** *(у веб-розробці) — неологізм, що описує модель роботи яка дозволяє реалізувати постійне HTTP-з'єднання і дає можливість відправляти дані з сервера за його ініціативою без додаткового запиту клієнта. Оскільки браузери та веб-сервери працюють по протоколу HTTP, який для подібних з'єднань не розрахований то розробники використовують різні реалізації. Streaming — в цьому випадку з'єднання постійно залишається відкритим і не закривається після кожної передачі даних. Цей підхід є складнішим і потребує спеціально програмного забезпечення. Реалізувати таку модель на стороні клієнта можна з допомогою JavaScript використовуючи AJAX або IFRAME. А на стороні сервера з допомогою, практично, будь-якого веб-сервера та мови програмування.*

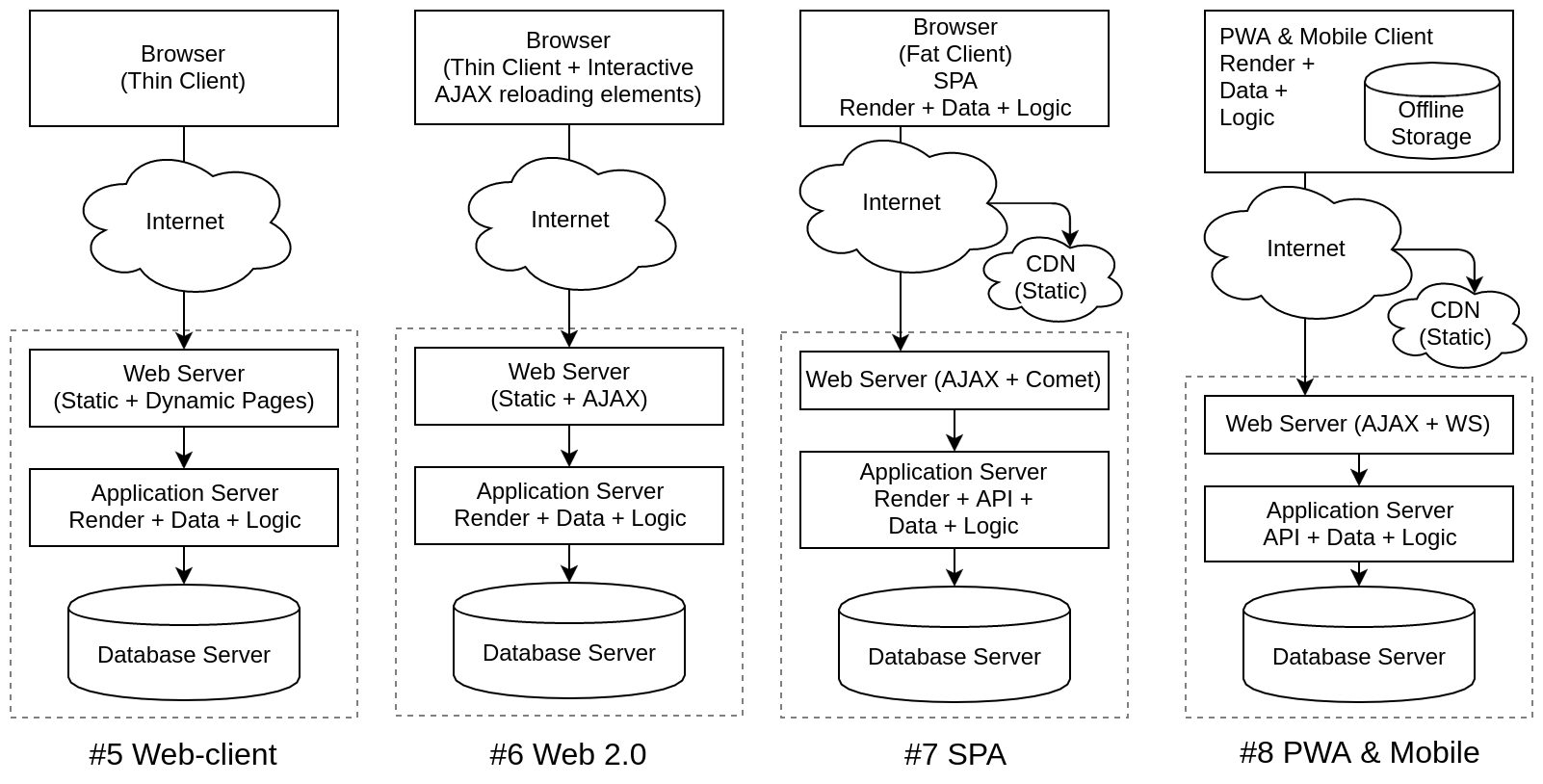
***CDN*** *- Сеть доставки (и дистрибуции) содержимого (англ. Content Delivery Network или Content Distribution Network, CDN) — географически распределённая сетевая инфраструктура, позволяющая оптимизировать доставку и дистрибуцию содержимого конечным пользователям в сети Интернет. Использование контент-провайдерами CDN способствует увеличению скорости загрузки интернет-пользователями аудио-, видео-, программного, игрового и других видов цифрового содержимого в точках присутствия сети CDN.*

***SPA***  *- Односторінковий додаток (англ. Single page application, SPA) - це веб-додаток або веб-сайт, який використовує єдиний HTML-документ як оболонку для всіх веб-сторінок і організуючий взаємодія з користувачем через динамічно підкачуємі HTML, CSS, JavaScript, зазвичай за допомогою AJAX.*

***PWA -*** *Progressive Web App are web applications that are regular web pages or websites, but can appear to the user like traditional applications or native mobile applications. The application type attempts to combine features offered by most modern browsers with the benefits of mobile experience.*

***WS*** *– Web services*





|  |  |  |
| --- | --- | --- |
| **Архітектура** | **Загальні особливості** | **Обмін** |
| #0 Data files | Всі зберігали свої дані в файлах різних невідомих форматів | Файли |
| #1 File-server | Двигуни БД, що працюють в одному процесі з додатком | Файли |
| #2 Local DB | СУБД виділилися в окремі процеси - сервера СУБД | IPC |
| #3 Client-Server | СУБД доступні по локалке різним типам АРМов (клієнтських робочих місць) | RPC |
| #4 Three-tier | Виділився шар сервера додатків, 3-ланкова, робота з віддалених офісів | RPC |
| #5 Web-client | З'явилися тонкі клієнти для доступу до даних з веб-браузера | HTTP |
| #6 Web 2.0 | Часткова інтерактивна перезавантаження елементів через AJAX, Comet, SSE | AJAX |
| #7 SPA/WebApp | Повнофункціональні браузерні додатки без перезавантаження сторінок | AJAX |
| #8 PWA & Mobile | Progressive Web Application і мобільні додатки | HTTPS |

Час існування для архітектур не вказано, тому, що і зараз в своїх нішах паралельно існують майже всі з перерахованих рішень. Для сучасних ІС переважна технологія все ще # 5, тобто звичайні веб сторінки з перезавантаженням по посиланнях (тонкий клієнт) і всією логікою на сервері. Для багатьох задач більшого й не потрібно. На передньому краю # 8, тут веб-додатки і мобільні додатки злилися архітектурно, хоч і мають безлічі відмінностей в технологіях і реалізації.

Історично першою з'явилася модель розподіленого представлення даних (модель сервер терміналів). Вона реалізовувалася на універсальній ЕОМ (мейнфрейми), що виступала в ролі сервера, з підключеними до неї алфавітно-цифровими терміналами. Користувачі виконували введення даних з клавіатури терміналу, які потім передавалися на мейнфрейм і там виконувалася їх обробка, включаючи формування «картинки» з результатами. Ця «картинка» і поверталася користувачеві на екран терміналу.

З появою персональних комп'ютерів і локальних мереж, була реалізована модель файлового сервера, який представляв доступ файлових ресурсів, в т.ч і до віддаленої базі даних. В цьому випадку виділений вузол мережі є файловим сервером, на якому розміщені файли бази даних. На клієнтах виконуються додатки, в яких поєднані компонент уявлення та прикладної компонент (СУБД і прикладна программма), що використовують підключену віддалену базу як локальний файл. Протоколи обміну при цьому представляють набір низькорівневих викликів операцій файлової системи.

Така модель показала свою неефективність з огляду на те, що при активній роботі з таблицями БД виникає велике навантаження на мережу. Частковим вирішенням цієї проблеми є підтримка тиражування (реплікації) таблиць і запитів. В цьому випадку, наприклад при зміні даних, оновлюється не вся таблиця, а тільки модифікована її частина.

З появою спеціалізованих СУБД з'явилася можливість реалізації іншої моделі доступу до віддаленої базі даних - моделі сервера баз даних. В цьому випадку ядро ​​СУБД функціонує на сервері, прикладна програма на клієнті, а протокол обміну забезпечується за допомогою мови SQL. Такий підхід в порівнянні з файловим сервером веде до зменшення завантаження мережі й уніфікації інтерфейсу "клієнт-сервер».

Однак, мережевий трафік залишається досить високим, крім того, як і раніше неможливо задовільний адміністрування додатків, оскільки в одній програмі поєднуються різні функції.

З розробкою і впровадженням на рівні серверів баз даних механізму збережених процедур з'явилася концепція активного сервера БД. У цьому випадку частина функцій прикладного компонента реалізовані у вигляді збережених процедур, що виконуються на стороні сервера. Решта прикладна логіка виконується на стороні клієнта. Протокол взаємодії - відповідний діалект мови SQL.

Переваги такого підходу очевидні:

• можливо централізоване адміністрування прикладних функцій;

• значне зниження мережевого трафіку (тому що передаються не SQL-запити, а виклики збережених процедур).

Основний недолік - обмеженість коштів розробки збережених процедур у порівнянні з мовами високого рівня.

Реалізація прикладного компонента на стороні сервера надає наступну модель - сервер додатків. Перенесення функцій прикладного компонента на сервер знижує вимоги до конфігурації клієнтів і спрощує адміністрування, але представляє підвищені вимоги до продуктивності, безпеки і надійності сервера.

**Базові технології web-розробки**

Якщо ви хочете навчитися своїми руками створювати сайти необхідно освоїти HTML, CSS, JavaScript, PHP і MySQL. Клієнтські технології (HTML, CSS, JavaScript), серверні (PHP, MySQL).

HTML (HyperText Markup Language) - це мова розмітки документа, що описує форму відображення інформації на екрані комп'ютера.

HTML впроваджує засоби для:

• создание структурованого документа Шляхом Позначення структурного складу тексту: заголовки, абзаци, списки, табліці, цитати та інше;

• Отримання информации Із Всесвітньої мережі через гіперпосілання;

• создание інтерактівніх форм;

• включення збережений, звуку, відео, та других об'єктів до тексту.

<html>

<head>

<meta charset = "utf-8">

<link rel = "stylesheet" type="text/css" href="myStyle.css">

<script type = "text/javascript" src = "myScript.js"></script>

<title>This is a simple page</title>

</head>

<body>

<header>

<h1>The head of the simple page</h1>

</header>

<section class = "main">

<div id = "my\_text">

<p>

This is a <span>first paragraph</span> of this page.

</p>

<p>

From here you can go to

<a href="http:\\anotherPage.com">Another

page</a>

</p>

</div>

<div id = "my-img">

<img src = "myImage.jpg" alt = "no image found">

</div>

</section>

<footer>

<h2>&copy; property of mine. 2017</h2>

</footer>

</body>

</html>

Каскадні таблиці стилів (CSS - Cascading Style Sheets) формальна мова опису зовнішнього вигляду документа, дозволяють істотно розширити можливості мови HTML за рахунок більш гнучкого управління форматуванням Web-сторінки. За допомогою каскадних таблиць стилів (CSS), можна задавати точні характеристики практично всіх елементів Web сторінки.

<font size="3">Текст</font>

<font style="font-size: 12pt">Текст</font>

<style type="text/css">

font { font-size: 12pt; }

</style>

<head>

<link rel="stylesheet" type="text/css"

href="style.css">

</head>

body {

text-align: center;

}

header, footer {

background-color: gray;

}

section.main {

font-size: 20px;

color: blue;

}

section.main a {

text-decoration: none;

color: skyblue;

font-weight: bold;

}

div#my-img img {

border: solid 1px black;

width: 400px;

height: 100px;

}

У Web-сторінок, створених з використанням HTML і CSS, є істотний недолік - вони є статичними, тобто не можуть змінюватися, реагуючи на дії користувача.

(Не зовсім так, з огляду на нові можливості HTML5 і CSS4, проте дозволяє реалізовувати тільки обмежений набір можливостей)

Основним інструментом роботи і динамічних змін на сторінці є DOM (Document Object Model) - об'єктна модель, використовувана для XML / HTML-документів.

Згідно DOM-моделі, документ є ієрархією, деревом. Кожен HTML-тег утворює вузол дерева з типом «елемент». Вкладені в нього теги стають дочірніми вузлами. Для представлення тексту створюються вузли з типом «текст».

JavaScript - це мова програмування, що дозволяє зробити Web-сторінку інтерактивною, тобто реагує на дії користувача. Впровадження в HTML програм на мові JavaScript дозволить "оживити" Web-сторінку, зробити її інтерактивною, або, іншими словами, змусити взаємодіяти з користувачем.

Структурно JavaScript можна представити у вигляді об'єднання трьох чітко помітних одна від одної частин:

• ядро ​​(ECMAScript),

• об'єктна модель браузера (Browser Object Model або BOM),

• об'єктна модель документа (Document Object Model або DOM).

Специфікація ECMAScript описує типи даних, інструкції, ключові і зарезервовані слова, оператори, об'єкти, регулярні вирази, не обмежуючи авторів похідних мов в розширенні їх новими складовими.

Об'єктна модель браузера - браузер-специфічна частина мови, що є прошарком між ядром і об'єктною моделлю документа. Основне призначення об'єктної моделі браузера - управління вікнами браузера і забезпечення їх взаємодії. Кожне з вікон браузера представляється об'єктом window, центральним об'єктом DOM.

Об'єктна модель документа - інтерфейс програмування додатків для HTML і XML-документів. Згідно DOM, документ може бути представлений у вигляді дерева об'єктів, що володіють рядом властивостей, які дозволяють виробляти з ним різні маніпуляції:

* генерація і додавання вузлів,
* отримання вузлів,
* зміна вузлів,
* зміна зв'язків між вузлами,
* видалення вузлів.

<script type="text/javascript">

document.write("Hello, world");

</script>

(особо востребованной является возможность событийной привязки скрипта к тем или иным действиям пользователя: on-click, on-change, …)

<a href="action.php" id="alertLink">

Make some action

</a>

window.onload = function() {

var linkWithAlert =

document.getElementById("alertLink");

linkWithAlert.onclick = function() {

return confirm(Are you sure that you want to

make this action?');

};

};

Ще більші можливості дає використання серверних технологій, серед яких може бути вибрана мова програмування PHP. Це найбільш поширений в даний час мова для написання серверних скриптів. Використовуючи його (або інші технології, що застосовуються для створення динамічних Web-сторінок), можна змінювати HTML-код, отриманий Web-браузером, в залежності від

• вводяться користувачем даних,

• типу і версії використовуваного Web-браузера

• та інших факторів.

Величезна кількість розширень і готових програмних продуктів, а також легкість освоєння мови зробили PHP дуже популярною мовою програмування для Інтернету. За допомогою PHP можна

• автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в визначені масиви;

• взаємодія з великою кількістю різних систем управління базами;

• автоматизована відправка HTTP-заголовків;

• робота з HTTP-авторизацією;

• робота з cookies і сесіями;

• робота з локальними і віддаленими файлами, сокетами;

• обробка форм.

На сьогоднішній день жоден великий портал не обходиться без використання баз даних. У Web-технологіях найчастіше застосовується швидка, безкоштовна і володіє великими можливостями система управління базами даних (СКБД) MySQL.

За допомогою MySQL можна

• ефективно додавати, змінювати і видаляти дані,

• отримувати потрібну інформацію за запитом.

• ефективне реалізововать розрахований на багато користувачів режим роботи

• Забезпечувати цілісність даних і отказоувтойчівость системи

PHP забезпечує ефективну підтримку баз даних і дозволяє працювати з MySQL, Microsoft SQL Server, Oracle, Sybase та ін. Знаючи і вміючи працювати з MySQL, легко перейти і до інших баз даних, якщо виникне така необхідність.

**Технологія CGI**

Спробуємо розібрати, що відбувається в загальному вигляді, коли ви вводите в адресному рядку браузера URL (Universal Resource Location, або в просторіччі - адреса)? Схема роботи наступна:

• Браузер відкриває з'єднання з сервером

• Браузер відправляє серверу запит на отримання сторінки

• Сервер формує відповідь (найчастіше - HTML-код) браузеру і закриває з'єднання

• Браузер обробляє HTML-код і відображає сторінку

Ще до того, як ви побачили на екрані запитану сторінку, з'єднання з сервером закрито, і він про вас забув. І коли ви введете інший (або той же самий) адреса, або оберіть пункт, або натиснете на кнопку HTML-форми - та ж схема повториться знову. (Виняток становить використання функцій кешування для прискорення роботи сайтів)

Розлянемо приклад html-форми:

<form method="GET" action="/cgi-bin/form\_handler.php">

Введіть Ваше им’я: <input type="text" name="name"><br>

<input type="submit" name="okbutton" value="OK">

</form>

Введіть Ваше им’я:   

Отже, що ж відбувається, коли ми натискаємо кнопку "OK"?

• Браузер переглядає елементи, що входять в форму і формує з них атрибутів name і value дані форми. Припустимо, введено ім'я Vasya. У цьому випадку дані форми – «name=Vasya&okbutton=OK»

• Браузер встановлює з'єднання з сервером, відправляє на сервер запит документа, зазначеного в атрибуті action тега <form>, використовуючи метод відправки даних, вказаний в атрибуті method (в даному випадку - GET), передаючи в запиті дані форми.

• Сервер аналізує отриманий запит, формує відповідь, відправляє його браузеру і закриває з'єднання (цей пункт розглядаємо далі детальніше)

• Браузер відображає отриманий від сервера документ

Веб-сервер вміє просто віддавати запитану сторінку. А для того, щоб можна було обробити ці дані за допомогою будь-якої програми і динамічно сформувати відповідь браузеру, і була винайдена технологія CGI (Common Gateway Interface)

Поглянемо на цей URL: http://www.example.com/cgi-bin/form\_handler.php. Перше припущення, яке можна зробити на його рахунок, зазвичай таке: сервер віддає вміст файлу form\_handler.php з каталогу cgi-bin. Однак, у випадку з технологією CGI справа йде по-іншому. Сервер запускає програму form\_handler.php і передає їй дані форми. Програма ж формує текст, який передається браузеру як відповідь на запит.

Програму form\_handler.php можна написати на будь-якій мові програмування, головне - дотримуватися в програмі стандарт CGI. Крім PHP широко використовуэться Java, Python, Ruby, C#, Node js. Кожен з них має свої сильні і слабкі сторони, тому розробник повинен зробити вибір виходячи з поставлених перед ним завдань.

Останнім часом PHP вибився в лідери за популярністю, чому є кілька причин:

• Простота. Мова дуже легко сприймається, особливо початківцями програмістами.

• Швидкість розробки. Завдяки простоті і інтуїтивної зрозумілості, PHP дозволяє дуже швидко створювати досить складні сайти.

• Наявність бібліотек. Існує величезна кількість готових прикладів і бібліотек класів. Створено та перевірено сотні бібліотек, істотно спрощують життя розробника.

• Підтримка. Практично будь-який сервер в Інтернеті підтримує PHP

• Безпека PHP дозволяє створювати дійсно захищені сайти завдяки встроненной підтримки шифрування даних при зберіганні і передачі.

• Комбінування html і php коду

**Історія розвитку PHP**

У 1994-му році, один програміст, на ім'я Rasmus Lerdorf, намучавшись з класичним перлової модулем CGI, вирішив написати кілька власних Perl-скриптів, щоб було простіше створювати власну домашню сторінку, і назвав все це справа Personal Home Page (PHP).

Через деякий час йому знадобилося обробляти форми, ну і для збільшення продуктивності все було переписано на C - так з'явився Personal Home Page / Forms Interpreter (PHP / FI) 2.0. Праці свої Расмус, слідуючи принципам Open Source, виклав на загальний огляд, і, в принципі, на певній кількості сайтів PHP / FI цілком успішно використовувався, хоча був досить примітивний.

У 1997-му на PHP / FI - в пошуках інструменту для зручного Веб-скриптинга - натрапили два інших програміста - Andi Gutmans і Zeev Suraski. Сама ідея їм сподобалася, але функціональність і швидкість роботи PHP / FI залишали бажати кращого, і Andi і Zeev зважилися переписати PHP з нуля.

Мова вийшов універсальний і потужний, і незабаром привернув увагу безлічі веб-розробників: до кінця 1998 року PHP3 використовувався на ~ 10% веб-серверів. Скромну назву "Personal Home Page" вже не дуже-то відповідало реальності, і назва була змінена на - в кращих Unix-традиціях - рекурсивне: PHP: Hypertext Preprocessor.

Однією з найсильніших сторін PHP 3.0 була можливість розширення ядра додатковими модулями. Згодом інтерфейс написання розширень привернув до PHP безліч сторонніх розробників, що працюють над своїми модулями, що дало PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. Велика кількість розробників призвело до швидкого розвитку мови і стрімкого зростання його популярності.

"Движок" PHP 4, названий Zend Engine, розроблявся зусиллями вже сформувався і з тих пір безперервно расрастающегося PHP community, і в 2000-му році вийшла 4-я версія PHP, що стала менш ніж через півроку стандартом для Веб-розробки під Unix (і не тільки): кожен поважаючий себе хостер надавав підтримку PHP.

PHP 4.0 мав ще кілька ключових нововведень, таких як підтримка сесій, буферизація висновку, безпечніші способи обробки вводиться користувачем, і декілька нових мовних конструкцій.

П'ята версія PHP була випущена розробниками 13 липня 2004 року. Зміни включають оновлення ядра Zend (Zend Engine 2), що істотно збільшило ефективність інтерпретатора. Введена підтримка мови розмітки XML. Повністю перероблені функції ООП, які стали багато в чому схожі з моделлю, використовуваною в Java. Зокрема, введена деструкція, відкриті, закриті і захищені члени і методи, остаточні члени і методи, інтерфейси і клонування об'єктів.

Шоста версія PHP розроблялася з жовтня 2006 року. Було зроблено безліч нововведень. Одним з основних нововведень повинна була стати підтримка Юнікоду. Однак у березні 2010 року розробка PHP6 була визнана безперспективною.

У 2014 році було проведено голосування, за результатами якого наступна версія отримала назву PHP 7. 3 грудня 2015 року було оголошено про вихід PHP версії 7.0.0. Нова версія грунтується на експериментальній гілки PHP, яка спочатку називалася phpng (PHP Next Generation - наступне покоління), і розроблялася з упором на збільшення продуктивності і зменшення споживання пам'яті. У новій версії додана можливість вказувати тип повертаються з функції даних, доданий контроль переданих типів для скалярних даних, а також нові оператори.

PHP - це мова програмування, що виконується на стороні сервера. На відміну від мови JavaScript PHP не залежить від програмного забезпечення клієнта і тому буде виконаний в будь-якому випадку.

Послідовність інструкцій (звана програмою або скриптом) виконується інтерпретатором мови PHP. Код програми може впроваджуватися в HTML-код. Обробка PHP-коду проводиться на сервері до того, як сторінка буде передана Web-браузеру. У підсумку Web-браузер отримає звичайний HTML-код.

Розглянемо остаточний приклад обробки форми введення призначених для користувача даних.

<html>

<head>

...

</head>

<body>

<?php if ($\_GET['name'] != “”) : ?>

<h1>Привіт <b>

<?php

echo $\_GET['name'];

?>

</b> !</h1>

<?php else : ?>

<form method="GET" action="form\_handler.php">

Введіть Ваше им’я:

<input type="text" name="name">

<input type="submit" name="okbutton" value="OK">

</form>

<?php endif; ?>

</body>

</html>

https://habrahabr.ru/post/326016/