

# Урок 2

Протокол HTTP

Основы PHP

# В уроке

- Что такое HTTP
- Архитектура клиент-серверного взаимодействия
- Архитектура протокола HTTP
- Установка веб-сервера на локальный компьютер
- Основы PHP
  - Базовый синтаксис
  - Переменные
  - Константы
  - Типы данных
  - Преобразование типов
  - Операторы и операции
  - Логические операции
  - Операции над строками
  - Приоритетность операторов

**Что происходит между тем, как вы  
набрали адрес сайта и увидели  
страницу на экране?**

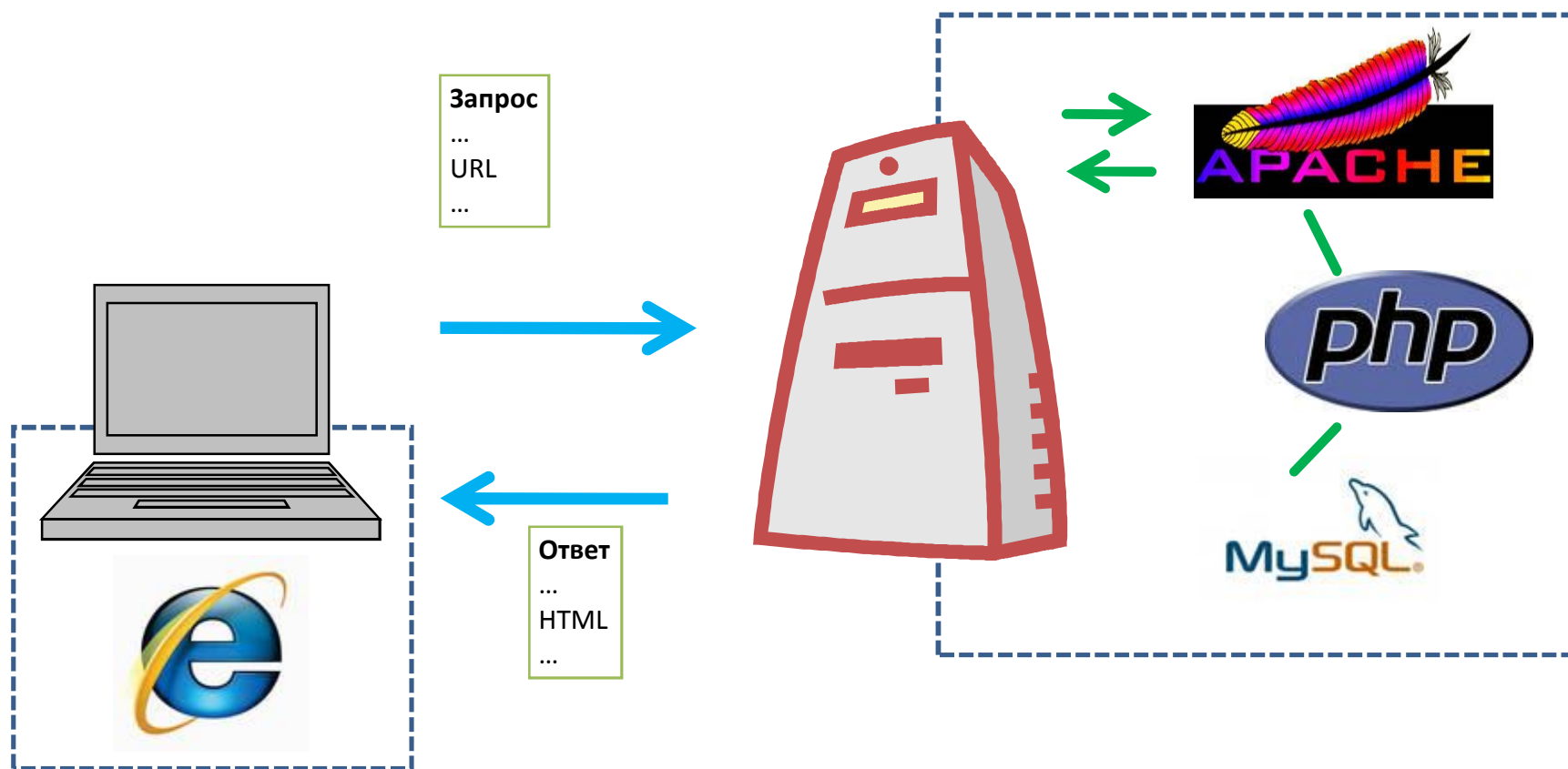
?

# Структура адреса сайта

<http://prog-school.ru/catalog>

- **http://**
  - Протокол взаимодействия с сервером
- **prog-school.ru**
  - Адрес сайта
- **/catalog**
  - Подзапрос к сайту

# Схема клиент-серверного взаимодействия между браузером и веб-сервером



# HTTP

- **HTTP** (сокращение от англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных.
- **Основой HTTP является технология «клиент-сервер»**, то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

# Структура HTTP-пакетов

- **Стартовая строка** — определяет тип сообщения;
- **Заголовки** — характеризуют тело сообщения, параметры передачи и прочие сведения;
- **Тело сообщения** — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

# Примеры стартовых строк HTTP-пакетов

- **HTTP-запрос**

**Вид:** **Метод** **URI** **HTTP/Версия**

- **Метод** — название запроса, одно слово заглавными буквами.
- **URI** определяет путь к запрашиваемому документу.
- **Версия** — пара разделённых точкой арабских цифр. Например: 1.1.

- **HTTP-ответ**

**Вид:** **HTTP/Версия** **КодСостояния** **Пояснение**

- **Версия** — пара разделённых точкой арабских цифр как в запросе.
- **КодСостояния** — три цифры. По коду состояния определяется дальнейшее содержимое сообщения и поведение клиента.
- **Пояснение** — текстовое короткое пояснение к коду ответа для пользователя. Никак не влияет на сообщение и является необязательным.



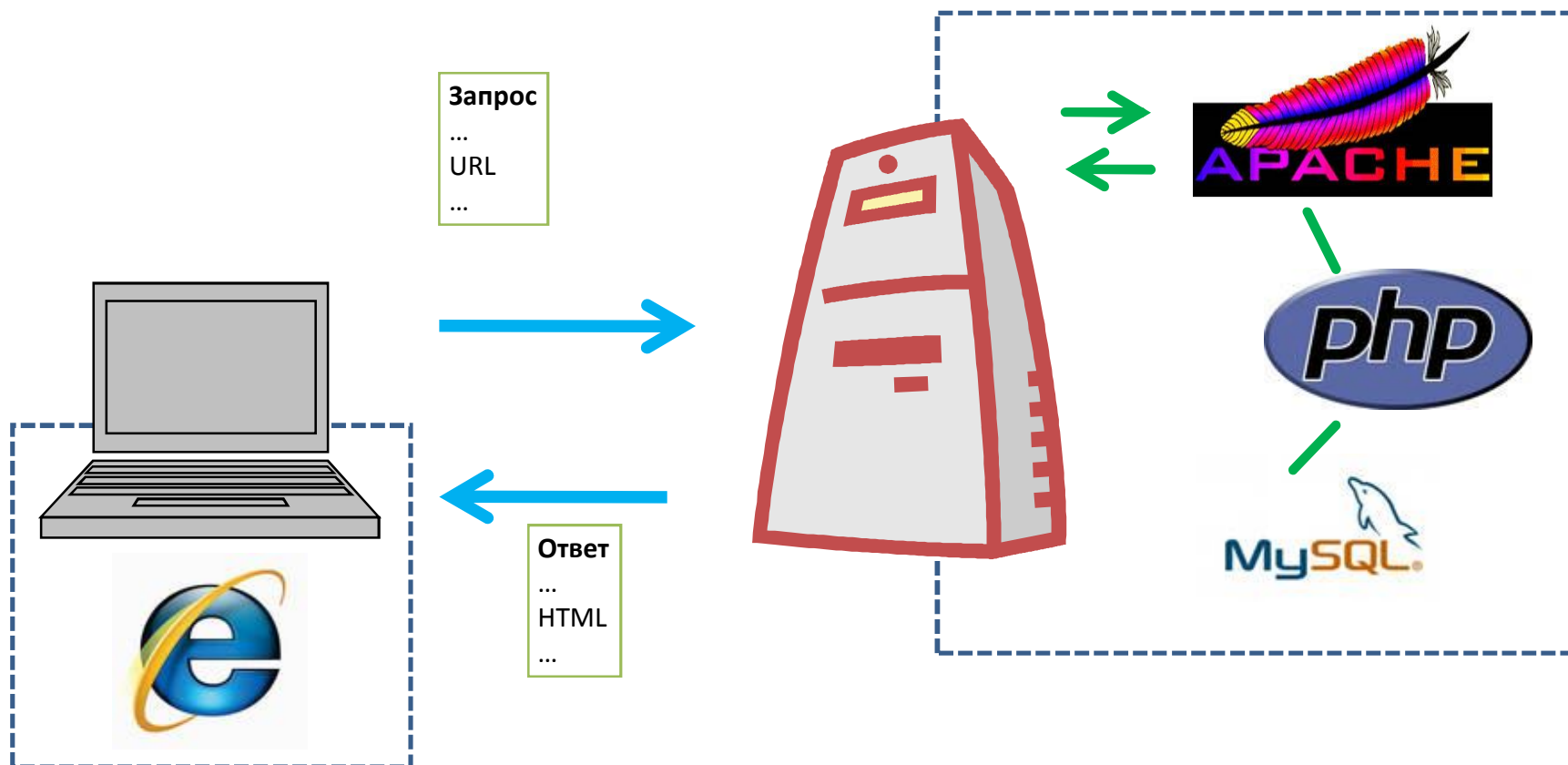
# Заголовки HTTP

- **Заголовки HTTP** (англ. HTTP Headers) — это строки в HTTP-сообщении, содержащие разделённую двоеточием пару параметр-значение. Заголовки должны отделяться от тела сообщения хотя бы одной пустой строкой.
- Примеры заголовков:
  - Server: Apache/2.2.11 (Win32) PHP/5.3.0
  - Last-Modified: Sat, 16 Jan 2010 21:16:42 GMT
  - Content-Type: text/plain; charset=windows-1251
  - Content-Language: ru

# Установка веб-сервера на локальный компьютер

- Перейдите на сайт <http://www.denwer.ru/>
- Скачайте наиболее свежую версию дистрибутива.
- Запустите инсталлятор и следуйте его инструкциям.
- Вспомогательное видео: <http://prog-school.ru/2010/02/kak-ustanovit-veb-server-i-napisat-pervyj-php-skript/>

# Еще раз о PHP



# Базовый синтаксис

```
<?php  
    echo "Hello!";  
?>
```

# Переменные

- Начинаются с «\$»
- Первый символ после «\$» - буква или знак «\_»
- В число допустимых символов входят заглавные и прописные латинские буквы, цифры, символ подчеркивания («\_»)
- По-русски можно, но **нельзя!**

# Комментарии

- `/* текст комментария */`
- `// текст комментария до конца строки`

# Константы

- После присваивания константе значения его уже нельзя изменить.
- Создаются с помощью функции `define()`:  
`define("INDEPENDENCEDAY", "4th July");`

# Типы данных

- РНР - свободно-типизированный язык.
- Типы данных РНР:
  - Целые
  - Действительные с плавающей точкой
  - Строковые
  - Булевские
  - Объекты
  - Массивы



# Целые числа

- Значения со знаком величиной до чуть более 2-х миллиардов
- Можно представить в:
  - Десятичной нотации
  - Восьмеричной нотации
  - Шестнадцатеричной нотации

# Числа с плавающей точкой

// Стандартная нотация с десятичной точкой

```
$my_float = 5.1;
```

// То же число в экспоненциальном

// представлении с плавающей точкой

```
$my_float = .051e2;
```

# Строки

- В двойных кавычках
- В одинарных кавычках

# Булевские переменные

- Два значения: истина, ложь
- true
- false

# Преобразование типов

- Используется для того, чтобы указать интерпретатору на необходимость работать со значением переменной одного типа так, как если бы она имела другой тип.
  - (string) – строка
  - (integer) или (int) – целое число
  - (double) – число с плавающей точкой
  - (boolean) или (bool) – булевская переменная (true/false)
- Пример:

```
$a = "432.123";  
echo((int)$a);
```

# Операторы и операции

- Операции
  - Арифметические
  - Сравнения
  - Над строками
  - Логические

# Арифметические операции

`$answer = 5 + 4; /* сложение */`

`$answer = $answer - 5; /* вычитание */`

`$answer = $answer / 2; /* деление */`

`$answer = $answer * $answer; /* умножение */`

`$answer = $answer % 3; /* остаток от деления */`

# Арифметические операции

- аргументы – целые числа, результат - целое число
- хотя бы один из аргументов – число с плавающей точкой, результат – число с плавающей точкой, даже если получается целое число.
- Пример:  $0.5 + 1.5 = 2.0$  , а не 2.



# Сокращенные версии арифметических операций

`$answer += 4;`

`$answer -= 5;`

`$answer /= 2;`

`$answer *= $answer;`

`$answer %= 3;`

# Инкремент декремент

- Инкремент – увеличение на единицу
  - $\$i++$
  - $++\$i$
- Декремент – уменьшение на единицу
  - $\$i--$
  - $--\$i$

# Операции сравнения

- Меньше <
  - Меньше или равно <=
  - Больше >
  - Больше или равно >=
  - Равно ==
  - Не равно !=
  - Идентично ===
  - Не идентично !==
- 
- Возвращают булевские значения true или false

# Операции над строками

## Конкатенация – объединение строк

```
$a = "Hello, ";
```

```
$b = "World";
```

```
$c = $a . $b;
```

```
echo($c);
```

Выведет: «Hello, World»

# Конкатенация

## Сокращенная запись

```
$a .= $b;
```

Пример:

```
$a = "Hello, ";
```

```
$b = "World";
```

```
$a .= $b;
```

```
echo($a);
```

Выведет: «Hello, World»

# Логические операции

Выполняются над булевскими операндами

Возвращают булевское значение

- И (and или &&)
- ИЛИ (or или || )
- НЕ (!)
- Исключающее ИЛИ (xor)

# Логические операции

## Примеры

```
$a = true;
```

```
$b = false;
```

```
$c = $a && $b;
```

```
$d = $a || $b;
```

```
$e = !$a;
```

```
$f = $a xor $b;
```

# Приоритетность операторов

`$c = $a++ + 2 <= --$b * 4 + $e && $f;`

???



# Приоритетность операторов

- Запоминать приоритеты – неправильно
- Использовать скобки – правильно
- Пример:  
$$\$c = (\$a++ + 2) \leq ((--\$b * 4) + (\$e \&\& \$f));$$

# Резюме

- Не зубрите справочную информацию, научитесь пользоваться гуглом
- Основы PHP – не самая интересная, но очень важная тема. Запаситесь терпением, скоро начнется практика!