Website design for artists (INTRO)

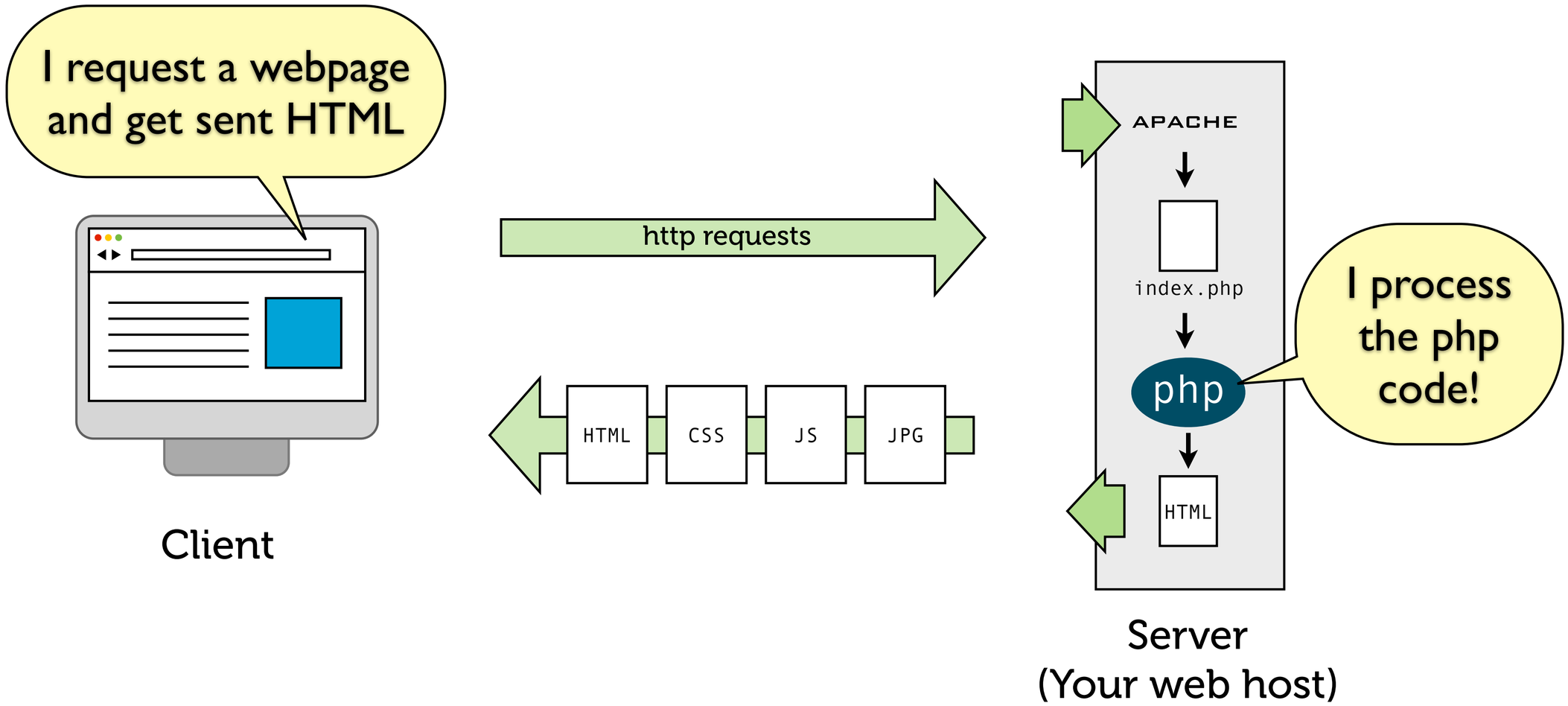
#7 Basic PHP

Today

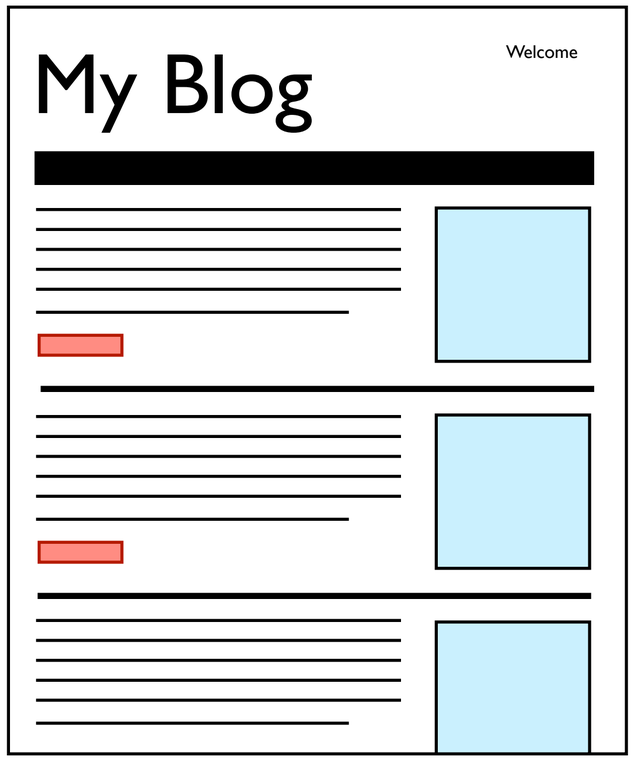
- What is PHP?
- Where did it come from?
- How does it work?
- What can be done with it?

What is this pay-aytch-pee?

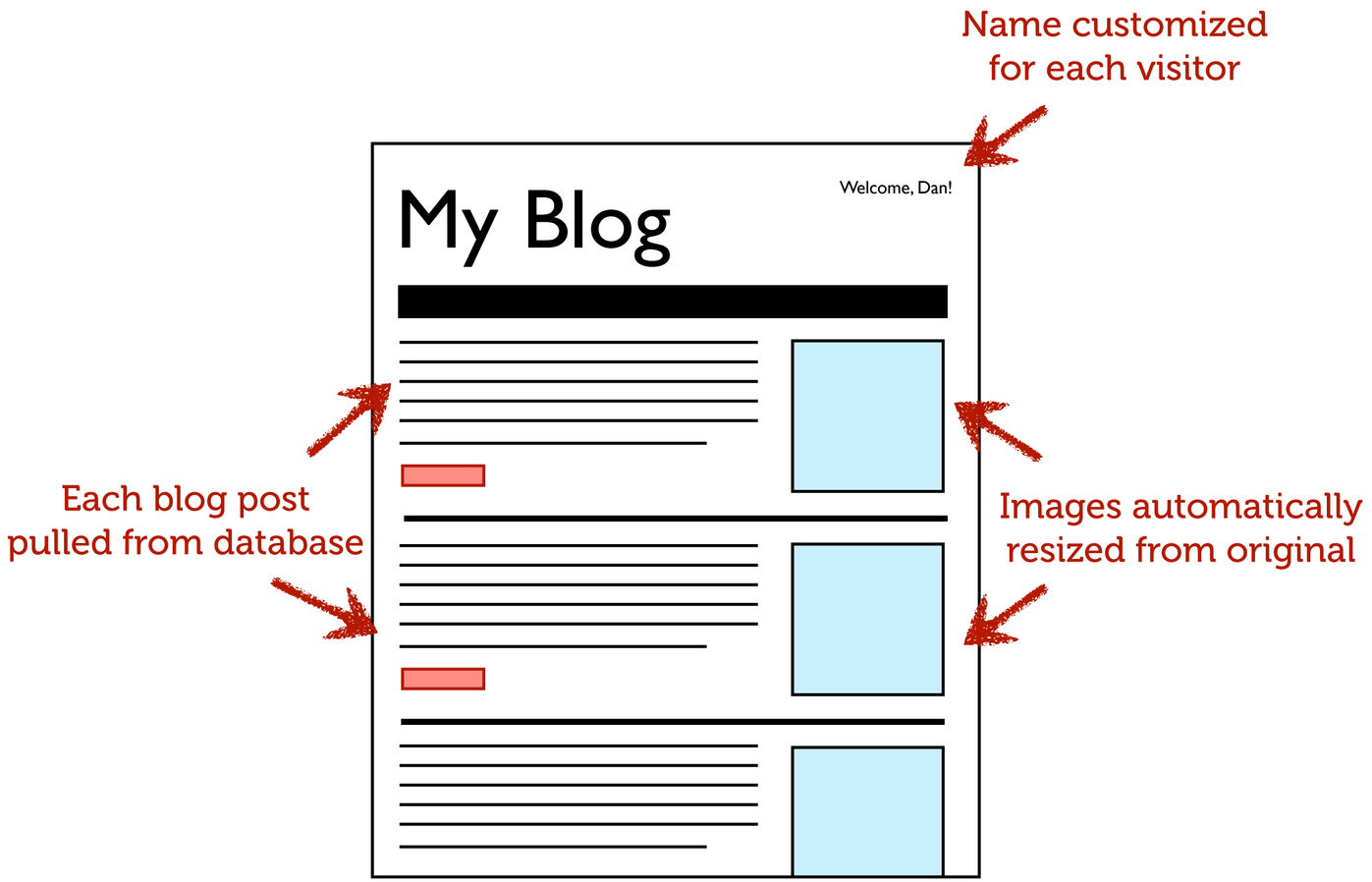
- A server-side scripting language specifically designed for web development
- Available as free software, and is pre-installed on most hosting platforms
- Very popular scripting language used on millions of websites
- Allows you to create “dynamic” web pages



Server-side scripting



index.html



index.php

Static vs. Dynamic page

A brief history

- 1994—Rasmus Ledorf writes pearl scripts for his webiste and names the system *Personal home page*

- *1995—Scripts rewritten, extended and released as PHP Tools v1.0*

- *1997—PHP development team formed and PHP 2.0 released*

- *1998—PHP 3.0, renamed to PHP: Hypertext Preprocessor*

- *Today—PHP 5.3*

Rasmus Lerdorf

How PHP works

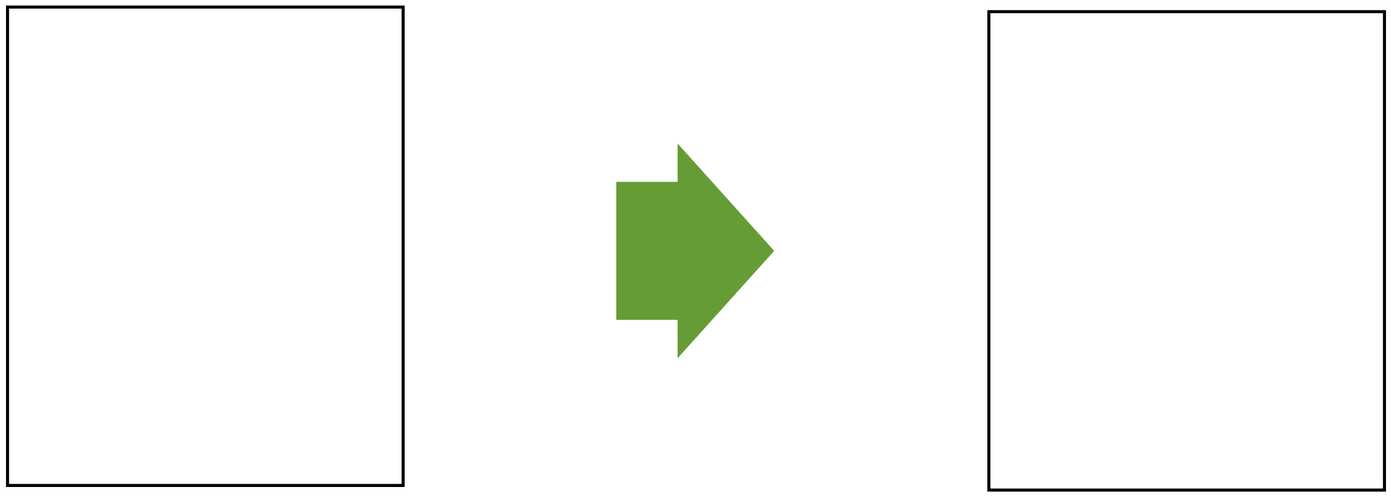
```
...
</head>
<body>
   <p><?php print 'Hello'; ?></p>
</body>
</html>
```

PHP code is embedded directly into HTML

```
...
</head>
<body>
  <p><?php print 'Hello, 5 + 5 = ' . (5 + 5); ?></p>
</body>
</html>
```

```
...
</head>
<body>
  <p>Hello, 5 + 5 = 10</p>
</body>
</html>
```

When the page is processed,
all php code is parsed and replaced



index.html

index.php

File extension must be changed to .php

code goes here

<?php ... ?>

PHP delimiters

CAUTION!

- Not following best practices and writing sloppy code will create security vulnerabilities. You will be hacked!
- Inefficient code will lead to slow downs and crashes!
- Be careful!

The PHP language

```
$name = 'Dan';
print 'Hello ' . $name;
```

Every instruction must end with a semicolon (;).
Forgetting even one will break the entire page.

```
// One-line comment

/* This is a multi-line
comment */
```

Comments (Good idea to use these)

Starts with dollar sign

Case sensitive

```
$php = 'awesome';
```

Must be letter or underscore

Variables

```php
$var = TRUE;
$var = 'Text';
$var = 1;
$var = 1.234;
$var = NULL;
```

Boolean

String

Integer

Float

No value

Some types

```php
$var = array(
    'name' => 'Dan',
    'age' => 28,
);
```

Key → 'name' ← Value

Arrays
(keys must be integer or string)

```php
$var = array(
    'name' => 'Dan',
    'age' => 28,
);

echo $var['name']; // Prints 'Dan'
```

Accessing array values

```php
$var = array(
  'name' => 'Dan',
  'age' => 28,
  'favorite colours' => array(
    'blue',
    'green',
  ),
);

print $var['favorite colours'][0]; // Prints 'blue'
```

Array values can be of any type, including arrays

$_GET
$_POST
$_SERVER
$_...

Some predefined variables

Operator

```php
<?php
if ($a > $b) {
   print "a is bigger than b";
}
?>
```

If

```php
<?php
if ($a > $b) {
  print 'a is bigger than b';
} else {
  print 'a is not bigger than b';
}
?>
```

else

```php
<?php
if ($a > $b) {
    print 'a is bigger than b';
} elseif ($a == $b) {
    print 'a is equal to b';
} else {
    print 'a is smaller than b';
}
?>
```

elseif

```php
<?php
$i = 1;
while ($i <= 10) {
   print $i;
   $i++;
}
? >
```

while loop

```php
<?php
$a = array(1, 2, 3, 17);

foreach ($a as $v) {
    echo "Current value of \$a: $v.\n";
}
?>
```

foreach

```php
<?php
$numbers = array(
    'one' => 1,
    'two' => 2,
    'three' => 3,
    'seventeen' => 17
);

foreach ($numbers as $k => $v) {
    print '$numbers[' . $k . '] => ' . $v . "\n";
}
?>
```

foreach (keys and values)

```php
<?php
include 'file.php';
?>
```

include

Operators

```php
$var = 'PHP is' . 'fun!';
$var .= 'Yeah!';
```

String operators

```
$a + $b
$a - $b
$a * $b
$a / $b
```

Addition

Substraction

Multiplication

Division

Arithmetic operators (yay math!)

```
$a = 5
$a += 3
$b .= 'text'
```

Assignment operators

Comparison operators

```
$a == $b
$a != $b
$a < $b
$a >= $b
```

← Equal

← Not equal

← Less than

← Greater than or equal to

Logical operators

```
$a  &&  $b   <-- And
$a  ||  $b   <-- Or
   !$a       <-- Not
```

Functions

```php
<?php
function add($arg1, $arg2) {
   return $arg1 + $arg2;
}

print add(1,5); // Prints 6
?>
```

User-defined functions

```php
<?php
function makecoffee($type = "cappuccino") {
    return "Making a cup of $type.\n";
}

print makecoffee();
print makecoffee(NULL);
print makecoffee("espresso");
?>
```

```
Making a cup of cappuccino.
Making a cup of .
Making a cup of espresso.
```

Default arguments

```php
<?php
function add_some_extra(&$string) {
    $string .= 'and something extra.';
}
$str = 'This is a string, ';
add_some_extra($str);
echo $str; // outputs 'This is a string, and something ext
ra.'
?>
```

Passing arguments “by reference”

Learn more at php.net