

# Programação II

Professores:

Claudio Esperança

Inês Dutra

## Aula 3

Professor:

*Inês Dutra*

Datas, horários e padrões

# Datas e Horários

- Datas e horários são muito importantes em PHP, pois muitas aplicações na web necessitam desta informação
- PHP pode tratar datas e horários em formatos especiais diferente do tratamento de strings

# Datas e Horários

- Datas e horários são armazenados no computador em um formato chamado *timestamp*
- O tipo de um *timestamp* é um inteiro ([Unix Timestamp](#)) que representa o número de segundos decorridos desde 1 de Janeiro de 1970 00:00:00 GMT ([Greenwich Mean Time](#))
- Este formato é muito conveniente para calcular intervalos de tempo através de diferenças entre *timestamps*
- Porém não é adequado para utilização pelo usuário

# Datas e Horários

- PHP pode converter *timestamps* para um formato adequado ao usuário e vice-versa
- Isto é feito através de predicados pré-definidos: *built-ins*

# Datas e Horários

```
$timestamp=date();
```

ou

```
$timestamp=strtotime("today");
```

Coloca na variável `$timestamp` a data corrente no formato entendido pelo computador

[Abrir editor](#)[Abrir navegador](#)

# Datas e Horários

- Built-ins:

- ▶ `date("format", $timestamp);`
- ▶ `strtotime("today");`

# Datas e Horários

```
$hoje = date("d/m/Y")  
echo $hoje;
```

## Resultado:

18/05/2006

[Abrir editor](#)

[Abrir navegador](#)

# Datas e horários: formatos

M	Jan
F	January
m	02, 12
n	2, 12
d	01, 16
j	1, 16
I	Fri
D	Friday
w	De 0 a 6 (dom a sab)
Y	2006
y	06
g	Hora entre 0 e 12, sem zeros: 1, 12
G	Hora entre 0 e 24, sem zeros: 1, 16
h	Hora entre 0 e 12, com zeros: 01, 12
H	Hora entre 0 e 24, com zeros: 01, 16
i	Minutos: 00, 59
s	Segundos: 00, 59
a	am, pm
A	AM, PM

# Datas e horários: formatos

- Pode-se trabalhar com datas e horários em formato muito similar à nossa linguagem
- Por exemplo, podemos converter datas e horários para o formato do computador da seguinte forma:

```
$timestamp=strtotime("today")
```

```
$timestamp=strtotime("now+24hours")
```

# Datas e horários

- Outros exemplos:

```
$data = strtotime("last saturday");  
$data = strtotime("8pm + 3 days");  
$data = strtotime("next year gmt");  
$data = strtotime("this 4am");  
$data = strtotime("2 weeks ago");
```

[Abrir editor](#)[Abrir navegador](#)

# Datas e horários

- **strtotime** reconhece os seguintes nomes e abreviações:
  - ▶ Nomes dos meses
  - ▶ Dias da semana
  - ▶ Unidades de tempo: **year**, **month**, **fortnight**, **week**, **day**, **hour**, **minute**, **second**, **am**, **pm**
  - ▶ Algumas palavras úteis: **ago**, **now**, **last**, **next**, **this**, **tomorrow**, **yesterday**

# Datas e horários

- Símbolos de mais e menos: +, -
- Números
- Horários dependendo da região:
  - ▶ **gmt**, Greenwich Mean Time
  - ▶ **pdt**, Pacific Daylight Time
  - ▶ **akst**, Alaskan Standard Time

# Datas e horários

- Assuma que queremos calcular o tempo decorrido de alguma data até hoje:

```
$hoje = date("today");  
$data = date("last Saturday");  
$intervalo = $hoje - $data;
```

O resultado é o tempo decorrido em segundos desde o último sábado até o horário corrente de hoje.

[Abrir editor](#)[Abrir navegador](#)

# Datas e horários

- Podemos também querer este tempo decorrido em horas:

```
$hoje = date("today");  
$data = date("last saturday");  
$intervalo = (($hoje - $data)/60)/60;
```

[Abrir editor](#)[Abrir navegador](#)

# Usando operadores relacionais

- Operadores relacionais são utilizados para comparar valor
- Por exemplo:
  - ▶ O preço de um produto é mais alto do que 100 reais?
  - ▶ A produção do produto A é maior que a do produto B?

# Usando operadores relacionais

<code>==</code>	Dois valores são iguais?
<code>&gt;</code>	Valor da esquerda é maior do que valor da direita?
<code>&gt;=</code>	Valor da esquerda é maior ou igual do que valor da direita?
<code>&lt;</code>	Valor da esquerda é menor do que valor da direita
<code>&lt;=</code>	Valor da esquerda é menor ou igual do que valor da direita?
<code>!=</code>	Dois valores são diferentes?
<code>&lt;&gt;</code>	Dois valores são diferentes?

# Usando operadores relacionais

- Geralmente utilizados em conjunto com comando **if-then-else**
- Exemplo:

```
if ($idade < 13)
    $status = "menor_de_idade";
else
    $status = "adulta";
```

- Este comando será visto mais tarde

# Comparação de padrões

- Padrões são também chamados de **expressões regulares**
- Por exemplo, podemos querer procurar nomes de arquivos que iniciem com a letra **c** e terminem com **.txt**
- **Provavelmente**, muitos de vocês já fizeram isto com o símbolo **\*** (coringa):

**dir c\*.txt ou ls c\*.txt**

# Comparação de padrões

- Padrões podem ser tão complexos quanto se queira
- Por exemplo, padrões são muito úteis para efetuar crítica de entrada em *forms*, onde o usuário digita um nome e o programa PHP verifica se o formato está correto

# Comparação de padrões

- Padrões consistem de **caracteres literais** e **caracteres especiais**
- **Caracteres literais** têm seu significado direto: por exemplo, o caracter **c** tem seu valor diretamente associado ao próprio caracter **c**
- Por outro lado, o caracter **\*** tem significado especial

# Caracteres especiais

Caracter	Significado	Ex	Match	Não match
^	Início de linha	^g	gato	O gato
\$	Fim de linha	o\$	gato	gatos
.	Qualquer caracter	..	Qualquer string com pelo menos 2 caracteres	a
?	Caracter precedente opcional	gat?o	gato, gao	grito

# Caracteres especiais

Caracter	Significado	Ex	Match	Não match
( )	Agrupa caracteres	g(at)o	gato	gao, go
[ ]	Agrupa caracteres opcionais	g[at]o	gto, gao	gato, go
-	Intervalo de caracteres	g[a-c]to	gato, gbto, gcto	gdto, gaato
+	Um ou mais dos caracteres precedentes	gato[1-3]+	gato1, gato11, gato131	gato, gato4

# Caracteres especiais

Caracter	Significado	Ex	Match	Não match
*	Zero ou mais dos caracteres precedentes	gato[1-3]*	gato, gato31	gato4
{ , }	Número de repetições	a{2,5}	aa,aaaaa	a,xx3
\	Caracter seguinte considerado como literal	20\*3	20*3	2053
(   )	Conjunto alternativo de caracteres	(gato gata)	gato, gata	grito

# Exemplos

- `^ [A-Z] . *`

Strings possíveis:

- `gato pulou do telhado`
- 

- Teste de CEP:

`^[0-9]{5}(-[0-9]{3})? $`

21920-030 ou 21920

[Abrir editor](#)

[Abrir navegador](#)

## Exemplos

- Qualquer string contendo **@** e que termina em **.com**

`^ .+@ .+ \ .com$`

**Ex:** `xpto@company . com`

# Comparação de padrões

- `ereg("padrao", "string")`
- Os argumentos de ereg também podem ser variáveis