

PHP a MySQL

Podmínky v PHP

Projekt
DUM

CZ.1.07/1.5.00/34.1009
VY_32_INOVACE_284

Ing. Karel Johanovský

Střední průmyslová škola Jihlava

2013



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Identifikační údaje

Projekt	<i>Inovace výuky prostřednictvím ICT</i>
<i>Číslo projektu</i>	<i>CZ.1.07/1.5.00/34.1009</i>
<i>Číslo DUM</i>	<i>VY_32_INOVACE_284</i>
<i>Autor</i>	<i>Ing. Karel Johanovský</i>
<i>Datum vytvoření</i>	<i>20. února 2014</i>
<i>Tematický celek</i>	<i>Programování a vývoj aplikací - PHP a MySQL</i>
Téma	<i>Podmínky v PHP</i>
<i>Anotace</i>	<i>Podpora výuky programování dynamických webů</i>
<i>Metodický pokyn</i>	<i>Prezentace s výkladem, časová náročnost 20 minut</i>
<i>Inovace</i>	<i>Podpora vjemu informací u žáka ve fázi expozice . a zejména ve fázi fixace získaných poznatků (dostupný materiál – možnost libovolného počtu opakování)</i>

1 Úvod

- Úvod

2 Větvení pomocí IF

- Klasický IF
- Neúplný IF
- IF - ELSEIF - ELSE

3 Větvení pomocí SWITCH

- Klasický SWITCH

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.
- Cílem této prezentace je ukázat základní řídicí struktury pro větvení programu.

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.
- Cílem této prezentace je ukázat základní řídicí struktury pro větvení programu.
 - IF - ELSE

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.
- Cílem této prezentace je ukázat základní řídicí struktury pro větvení programu.
 - IF - ELSE
 - Neúplný IF

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.
- Cílem této prezentace je ukázat základní řídicí struktury pro větvení programu.
 - IF - ELSE
 - Neúplný IF
 - IF - ELSEIF - ELSE

- Jazyk PHP podobně jako většina ostatních programovacích jazyků podporuje větvení programu.
- Cílem této prezentace je ukázat základní řídicí struktury pro větvení programu.
 - IF - ELSE
 - Neúplný IF
 - IF - ELSEIF - ELSE
 - SWITCH

- Klasické větvení

Klasický IF

- Klasické větvení
- Klíčové slovo IF

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }
 - Obsahuje co se má stát když JE podmínka splněna

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }
 - Obsahuje co se má stát když JE podmínka splněna
- Klíčové slovo ELSE

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }
 - Obsahuje co se má stát když JE podmínka splněna
- Klíčové slovo ELSE
- Nazávěr blok ohraničený { }

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }
 - Obsahuje co se má stát když JE podmínka splněna
- Klíčové slovo ELSE
- Nazávěr blok ohraničený { }
 - obsahuje co se má stát když NENÍ podmínka splněna

Klasický IF

- Klasické větvení
- Klíčové slovo IF
- Poté podmínka ohraničená ()
 - Musí být vyhodnotitelná jako (splněno / nesplněno)
- Následuje blok ohraničený { }
- Obsahuje co se má stát když JE podmínka splněna
- Klíčové slovo ELSE
- Nazávěr blok ohraničený { }
 - obsahuje co se má stát když NENÍ podmínka splněna

Klasický IF

```
IF (Podmínka) {  
    Co se má stát když je splněno  
} ELSE {  
    Co se má stát když není splněno  
}
```

Alternativní IF

- PHP podporuje i alternativní zápis příkazu IF.

- PHP podporuje i alternativní zápis příkazu IF.

Alternativní IF

```
IF (Podmínka) :  
    Co se má stát když je splněno  
ELSE :  
    Co se má stát když není splněno  
ENDIF;
```


Neúplný IF

Neúplný IF

- Stejně jako klasický IF s tím že chybí větev ELSE

Neúplný IF

- Stejně jako klasický IF s tím že chybí větev ELSE
- Pokud je splněna podmínka, provede se to co je uvedeno v { }

Neúplný IF

- Stejně jako klasický IF s tím že chybí větev ELSE
- Pokud je splněna podmínka, provede se to co je uvedeno v { }

Neúplný IF

```
IF (Podmínka) {  
    Co se má stát když je splněno  
}
```

Neúplný IF

- Stejně jako klasický IF s tím že chybí větev ELSE
- Pokud je splněna podmínka, provede se to co je uvedeno v { }

Neúplný IF

```
IF (Podmínka) {  
    Co se má stát když je splněno  
}
```

Alternativní neúplný IF

```
IF (Podmínka) :  
    Co se stát když je splněno  
ENDIF;
```

IF - ELSEIF - ELSE

IF - ELSEIF - ELSE

- Postupně se testují podmínky, pokud je nějaká vyhodnocena jako splněno, další už se netestuje.

IF - ELSEIF - ELSE

- Postupně se testují podmínky, pokud je nějaká vyhodnocena jako splněno, další už se netestuje.
- Větví ELSEIF může být i více.

IF - ELSEIF - ELSE

- Postupně se testují podmínky, pokud je nějaká vyhodnocena jako splněno, další už se netestuje.
- Větví ELSEIF může být i více.
- Větev ELSE je nepovinná a provede se pokud není splněna ani jedna podmínka.

IF - ELSEIF - ELSE

- Postupně se testují podmínky, pokud je nějaká vyhodnocena jako splněno, další už se netestuje.
- Větví ELSEIF může být i více.
- Větev ELSE je nepovinná a provede se pokud není splněna ani jedna podmínka.

IF - ELSEIF - ELSE

```
IF (Podmínka 1) {  
    Co se má stát když je splněna podmínka 1  
} ELSEIF (Podmínka 2) {  
    Co se má stát když je splněna podmínka 2  
} ELSEIF (Podmínka 3) {  
    Co se má stát když je splněna podmínka 3  
} ELSE {  
    Co se má stát když není splněna ani jedna  
}
```

IF - ELSEIF - ELSE Alternativně

IF - ELSEIF - ELSE

IF (Podmínka 1) :

Co se má stát když je splněna podmínka 1

ELSEIF (Podmínka 2) :

Co se má stát když je splněna podmínka 2

ELSEIF (Podmínka 3) :

Co se má stát když je splněna podmínka 3

ELSE :

Co se má stát když není splněna ani jedna

ENDIF;

Klasický SWITCH

Klasický SWITCH

- Větvení na více cest, na základě konkrétních hodnot určité proměnné.

Klasický SWITCH

- Větvení na více cest, na základě konkrétních hodnot určité proměnné.
- Ukončit příkazem BREAK, jinak pokračuje v provádění dál do konce switch.

Klasický SWITCH

- Větvení na více cest, na základě konkrétních hodnot určité proměnné.
- Ukončit příkazem BREAK, jinak pokračuje v provádění dál do konce switch.

SWITCH

```
SWITCH ($promenna) {  
    CASE hodnota 1:  
        Co se ma udelat pokud ma promenna hodnotu 1  
        BREAK;  
    CASE hodnota 2:  
        Co se ma udelat pokud ma promenna hodnotu 2  
        BREAK;  
    DEFAULT:  
        Co se ma udelat pokud nema ani jednu z vyjmenovanych  
hodnot  
        BREAK;  
}
```


- Představili jsme si příkazy pro větvení program v PHP.

- Představili jsme si příkazy pro větvení program v PHP.
- Ukázali jsme si příkazy IF - ELSE, Neúplný IF, IF - ELSEIF - ELSE a SWITCH.



T. Converse, J. Park, C. Morgan.

PHP5 and MySQL Bible

Wiley Publishing, Inc., 2004.

ISBN 0-7645-5746-7

- Tento materiál je určen pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.
- Všechna neocitovaná autorská díla jsou dílem autora.