**switch – case struktura**

**00:18:37**

Jedinica: 6 od 19

**+Rezime**

U ovoj lekciji radićemo naredbu switch i objasniti switch-case strukturu.

**Switch naredba**

if..elseif..else način uslovljavanja izvršenja nekog bloka koda je dobar, ali ne naročito elegantan, kada dođe do veće količine elseif alternativa pa se za slučajeve gde ima više alternativa koristi nešto preglednija switch case struktura. Drugim rečima, moglo bi se reći da je ova struktura idealna za testiranje neke promenljive ili izjave koja može imati više očekivanih vrednosti. Uzmimo, na primer, potrebu da se u kodu testira promenljiva koja bi trebalo da sadrži naziv dana u nedelji. Jasno je da možemo očekivati samo sedam vrednosti.

Grafički prikaz switch case strukture bi se mogao predstaviti na sledeći način:



*6.1. switch šema*

Pogledajmo sada sintaksu:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | switch (izraz) {    case vrednost1:          //Blok koda koji će se izvršiti ukoliko se izraz i vrednost 1 poklapaju      break; //Ključna reč kojom se prekida izvršavanje koda u switch strukturi    case vrednost2:       //Blok koda koji će se izvršiti ukoliko se izraz i vrednost 2 poklapaju      break;    default:          //Blok koda koji će biti izvršen ukoliko nema drugih poklapanja  } |

Kod *switch* strukture prvo se izračunava izraz, a zatim se dobijena vrednost poredi sa vrednostima navedenim u **case** oznakama. Ukoliko je vrednost izraza jednaka nekoj od vrednosti iz **case** oznake, izvršava se grupa naredbi koja sledi tu oznaku. Ukoliko je vrednost izraza različita od svih navedenih, izvršava se grupa naredbi nakon **default**oznake.

Ovu strukturu je jednostavno zameniti if..elseif..else strukturom, ali u nekim slučajevima switch..case je praktičniji i pregledniji. Na primer, u if-else strukturi morate prilikom svake provere da smeštate promenljive i uslovni izraz, dok je kod switch-case to nepotrebno. Kod switch-case, sa druge strane, ne možete da menjate promenljivu, već samo izraze nad njom. Osim toga, switch-case, iako dosta pregledan, nije baš podoban za velike uslovne blokove. Obično se u njega smeštaju ili kratke uslovljene akcije ili funkcije.

Pogledajmo primer:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | $x = 2;  switch( $x ){      case 1:          echo "x is one";          break;      case 2:          echo "x is two";          break;      default:          echo "x is not one either two";  } |

U primeru smo prvo inicijalizovali promenljivu $x, a zatim je postavili u switch strukturu. Switch struktura podrazumeva blokove uslova, obavezno oivičene vitičastom zagradom. Svaki uslov započinje ključnom rečju **case**, kojoj sledi uslovna vrednost ili uslov i dvotačka koja označava početak bloka. Ovaj blok možete i ne morate smestiti u vitičaste zagrade. Za razliku od standardnih uslovnih blokova, u slučaju **case** bloka, biće izvršen kompletan uslovljeni blok (a ne samo prvi red). Štaviše, ukoliko se uslov u nekom trenutku (slučaju) ispuni, biće izvršene sve naredne linije, sve dok program ne dobije direktivu da napusti kompletnu strukturu ili se struktura ne završi.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | $x = 1;  switch( $x ){      case 1:          echo "x is one";      case 2:          echo "x is two";      default:          echo "x is not one either two";  } |

Ovaj kod će u PHP-u kao rezultat dati sve tri echo naredbe, upravo zbog toga što nigde ne postoji izlaz iz strukture. To znači da je podrazumevani deo svakog case bloka i naredba break (bezuslovno napuštanje strukture). Ipak, postoje slučajevi kada programer namerno ne želi da postavi naredbu break. Ovo su slučajevi u kojima za dve ili više vrednosti izraza treba izvršiti isti blok koda.

Pogledajmo primer koji ovo ilustruje:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | $x = 1;  switch( $x ){      case 1:      case 2:          echo "x is one or two";          break;      case 3:      case 4:          echo "x is three or four";          break;      default:          echo "x is not either one, either two,either three, either four";  } |

Što se switch-case strukture tiče, PHP je, u odnosu na druge jezike, veoma fleksibilan. Dok su u drugim jezicima ove strukture najčešće ograničene samo na celobrojne vrednosti, PHP u strukturu može prihvatiti bilo koju vrednost i zatim je korektno obraditi kroz „slučajeve”.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | //$d=3;  $d="Wed";  switch ($d)  {  case "Mon":  case 1:    echo "Today is Monday";    break;  case "Tue":  case 2:    echo "Today is Tuesday";    break;  case "Wed":  case 3:    echo "Today is Wednesday";    break;  case "Thu":  case 4:    echo "Today is Thursday";    break;  case "Fri":  case 5:    echo "Today is Friday";    break;  case "Sat":  case 6:    echo "Today is Saturday";    break;  case "Sun":  case 7:    echo "Today is Sunday";    break;  default:    echo "Wonder which day is this ?";  } |

Osim skalarnih vrednosti, slučajevi mogu prihvatiti i kompletne izraze. Na primer, možemo proveriti da li je neka vrednost jednaka, veća ili manja od ulazne:

case: $a < 1 ...

Čak, možemo proveravati i uslove koji su van konteksta:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | $x = 7;  switch(true)  {       case $x%2 == 0;          echo "x is divisible by 2";          break;       case $x%3 == 0:          echo "x is not divisible by 2, but it is divisible by 3";          break;      default:          echo "x is not divisible by 2, either 3";  } |

U case uslov možemo smestiti i funkciju:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | $x = false;  switch(true)  {       case is\_int($x):          echo "x is a integer";          break;       case is\_string($x):          echo "x is a string";          break;      case is\_bool($x):          echo "x is a boolean";          break;  } |

Kada je u pitanju case izraz, možete se služiti promenljivama tipa integer, double i string, dok se nizovi i objekti ne mogu prihvatiti kao ulazne promenljive ukoliko nisu naznačeni kao prostiji tipovi.

**Više o naredbi *break***

U svim dosadašnjim primerima mogli ste uočiti naredbu ***break***na kraju svakog *case* bloka. Sledećim primerom objasnićemo značenje, odnosno osnovnu ulogu break  naredbe u okviru switch – case blokova naredbi.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | $category = "news";      switch ( $category)    {        case "news" :            echo ' <p> News from the world ... </p> ';            break;        case "politics" :            echo ' <p> Politics... </p> ';            break;        case "sport" :            echo ' <p> Last sport results ... </p> ';            break;        default :             echo ' <p> Welcome... </p> ';    } |

Prikazani primer je standardnih okvira i sintakse koju smo već objasnili. Svakako, podrazumeva se da je promenljiva $category u prethodnom delu koda već deklarisana, što smo i uradili na samom početku php skripte:

$category= "news";

Naš program bi na izlazu ispisivao:

*News from the world...*

Ovakav rad programa i switch–case strukture ostvaruje se upravo radom **break** naredbe. Evo o čemu je reč:

Prilikom izvršavanja, program vrši proveru koji od case blokova zadovoljava vrednost promenljive $kategorija. Kada je pronađe, nailazi na naredbu break i zaustavlja dalje izvršavanje programa. Kada bismo izostavili break naredbe iz predhodnog koda, naš php skript bi imao sledeću strukturu:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | $category= "news";      switch ( $category )    {        case "news" :            echo ' <p> News from the world...</p> ';        case "politics" :            echo ' <p> Politics... </p> ';        case "sport" :            echo ' <p> Last sport results ... </p> ';        default :             echo ' <p> Welcome ... </p> ';    } |

Izostavljanjem breaka u okviru case blokova, naš program izvršava sve case blokove zanemarujući switch uslovnu promenljivu. Ispis na izlazu nakon izvršenja biće:

*News from the world...  
Politics...  
Last sport results ...  
Welcome ...*

**Naredba *default***

Na kraju svake switch-case strukture nalazi se naredba **default**. Kada pokrenemo program, PHP pokušava da pronađe neki od **case** blokova koji zadovoljava unapred utvrđeni uslov; u slučaju da nijedan od ponuđenih uslovnih naredbi ne zadovoljava predviđene kriterijume, dolazi do izvršavanja upravo **default** naredbe. Default naredba je, najkraće rečeno, komanda koja se izvršava kada nijedna druga nije zadovoljavajuća. Evo i konkretnog primera:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $i = 5;  // sample  switch ( $i ) {      case 0:          echo "i is equal 0";          break;      case 1:          echo "i is equal 1";          break;      case 2:          echo "i is equal 2";          break;      default:         echo "i is not equal 0, 1 either 2";  } |

Nakon izvršenja programa, na izlazu dobijamo ispis:

*i is not equal 0, 1 either 2*

… što upravo predstavlja našu default naredbu.

Obratite pažnju na to da pozicija slučaja ne utiče na tok strukture. Sledeći primer će dati rezultat isti kao i prethodni, iako je redosled slučajeva drugačiji.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | $i = 5;  // sample  switch ( $i ) {      default:         echo "i is not equal 0, 1 either 2";       break;      case 1:          echo "i is equal 1";          break;      case 0:          echo "i is equal 0";          break;      case 2:          echo "i is equal 2";          break;  } |

|  |  |  |
| --- | --- | --- |
| **Napomena**    Treba imati u vidu da je verzija PHP 5.7 dozvoljavala upotrebu višestrukih default blokova, gde bi se bez problema izvršavao samo poslednji default blok. Pogledajmo to na primeru:     |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <?php    $x = 4;    switch($x){         case 1:                     echo 'Case 1';            break;           case 2:                     echo 'Case 2';            break;           case 3:                     echo 'Case 3';            break;           default:                     echo 'Default 1';            break;           default:                     echo 'Default 2';            break;  } |       Rezultat na strani bi izgledao ovako:    Default 2      Verzija PHP7 ukida ovakvo ponašanje i pri pojavi višestrukih default blokova generiše grešku. |

**Izostavljanje break naredbe u okviru case blokova rezultira:**

 Izvršavanjem svih case blokova koji slede nakon poklapanja uslova

 Zaustavljanjem programa

 Ispisivanjem sintaksičke greške

**Default naredba/slučaj je najkraće rečeno:**

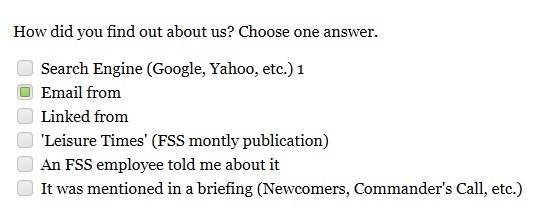
 Slučaj koji se izvršava kada nijedan uslov nije ispunjen

 Slučaj koji se u svakom slučaju izvršava, čak i ako je neki od uslova ispunjen

 Slučaj koja se nikada neće izvršiti

**Primer 1:**

Često se na sajtovima srećemo sa delom popunjavanja upitnika „Kako ste saznali za nas". Upravo ovi upitnici mogu funkcionisati preko switch–case struktura.



*6.2. Primer forme*„*Kako ste nas pronašli?"* – Na formi se se nalaze radio button

Svakako, sam izgled i odabir načina koji se povezuje sa case blokovima pripada naprednijim kursevima, i iz tog razloga nećemo kreirati HTML stranice sa padajućim menijima koji sadrže vrednosti naše promenljive za switch uslov. Objasnićemo samo switch–case deo php skripta.

Kada postavimo da neka naša promenljiva **$nacin** koju smo deklarisali sadrži vrednosti a, b, c ... kojima se dalje služimo kao odabranim poljima na stranici, možemo da koristimo sledeći kod:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $way = "b";    switch ($way){        case "a":            echo '<p> Regular customer ... </p>';            break;        case "b":            echo '<p> Customer has found us over the TV commercial.... </p>';            break;        case "c":            echo '<p> Recommendation... </p>';            break;        default :             echo '<p> We don't know how customer found us...</p>';    } |

Da bismo uočili rad programa, dodelili smo promenljivoj $nacin pre switch–case strukture vrednost „b" kako bismo simulirali odabir načina opredeljenja mušterije. Nakon izvršenja programa, na izlazu će biti ispisano:

Customer has found us over the TV commercial...

**Primer 2:**

Iz primera u nastavku možemo zaključiti da u okviru switch izraza možemo vršiti i operacije manjeg stepena kao što je, na primer, operacija uvećavanja:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <?php  $a = 0;    switch(++$a) {      case 3 :          echo 3;      break;      case 2:          echo 2;      break;      case 1:          echo 1;      break;      default :          echo "No matching...";      break;  }  ?> |

Nakon izvršavanja programa, na izlazu ćemo dobiti rezultat:

*1*

Što potvrđuje da je operacija u okviru switcha izvršena, pa je promenljiva koja je na početku imala vrednost 0, nakon prolaska kroz switch–case dobila vrednost 1.

**Vežba 1**

***Problem:***

Potrebno je napraviti kod koji će sadržati pet konstanti za radne dane nedelje (MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY).

Takođe, kod treba da sadrži deo koji proverava konstantu (dan nedelje) i na osnovu nje prikazuje opis dana putem switch-case strukture. Ukoliko je ponedeljak, opis je „prvi dan nedelje" itd. Potrebno je izvršiti testiranje kreiranog koda.

***Rešenje:***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | <?php    define( "MONDAY", 1 );    define( "TUESDAY", 2 );    define( "WEDNESDAY", 3 );    define( "THURSDAY", 4 );    define( "FRIDAY", 5 );      $day = MONDAY;      switch( $day ){      case MONDAY:          $description = "First day of week";          break;      case TUESDAY:          $description = "Second day of week";          break;      case WEDNESDAY:          $description = "Third day of week";          break;      case THURSDAY:          $description = "Forth day of week";          break;      case FRIDAY:          $description = "Fifth day of week";          break;      default:          $description = "no description";      break;    }    echo $description;  ?> |

Prvi zahtev u ovom zadatku je kreiranje konstanti. Podsetimo se da se konstante kreiraju upotrebom funkcije define(). Prvi parametar ove funkcije je naziv konstante, dok je drugi njena vrednost.

Takođe, definišemo i vrednost za promenljivu $day, kojom određujemo dan u nedelji. Zatim kreiramo switch i unutar običnih zagrada postavljamo promenljivu koju želimo da testiramo. Ova promenljiva biće upoređivana sa vrednostima koje su definisane nakon case, sve dok se ne ostvari poklapanje. Ukoliko se poklapanje nikada ne ostvari, izvršiće se default blok koda.

Na samom kraju dovoljno je da uz pomoć echo naredbe prikažemo vrednost za $description.

**Vežba 2**

***Problem:***

Potrebno je napraviti kod koji će na osnovu promenljivih $operator, $operand1 i $operand2 izvršiti računsku operaciju i prikazati rezultat.

Potrebno je upotrebiti switch-case strukturu u tu svrhu.

***Rešenje:***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <?php  $operator = "-";  $operand1 = 2;  $operand2 = 3;    switch($operator)          {              case "+":                  $result = $operand1 + $operand2;                  break;              case "-":                  $result = $operand1 - $operand2;                  break;              case "\*":                  $result = $operand1 \* $operand2;                  break;              case "/":                  $result = $operand1 / $operand2;                  break;              default:                  $result = "Unknown result.";          }  echo $result;  ?> |

Rešenje ovog zadatka je po svojoj strukturi veoma slično prethodnom. Najpre smo kreirali tri promenljive, kao što je to zahtevano u postavci, i odmah im dodelili vrednosti. Sada je potrebno da kroz switch proverimo kakvu vrednost ima promenljiva $operator, da bismo na osnovu toga mogli da izvršimo odgovarajuću računsku operaciju. Za svaki od potencijalnih izraza rezultat smeštamo u promenljivu $result.

Ukoliko je $operator string koji ne odgovara nijednoj vrednosti koja je definisana u switchu, kao rezultat se uzima string „Unknown result“.

**Vežba 3**

***Problem:***

Potrebno je napraviti program koji će obaveštavati korisnika o aktivnostima prilikom rada na kursu. Proverava se broj ulazaka u nastavnu jedinicu i na osnovu toga izdaju obaveštenja o početku učenja, testu po modulu i završetku kursa.

Koristiti switch–case strukturu. Definisati promenljivu za broj ulazaka i dodeliti joj vrednost 0 kako bi se omogućilo brojanje ulazaka u daljem kodu.

***Rešenje:***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?php  $numberOfEntries = 0;    switch ($numberOfEntries){      case 0 :          echo ' Welcome! You can start with learning...';          break;      case 20 :          echo ' It is time for test...';          break;      case 30 :          echo ' You have developed a condition for completion of the course....';          break;  }  ?> |

Na samom početku kreiramo promenljivu $numberOfEntries, koja će predstavlljati broj ulazaka na platformu za učenje. Za korisnika želimo da generišemo različite poruke za tri broja ulazaka i to: prvi put, nakon 20 i nakon 30 poseta. Koristimo se switch strukturom kao i kod prethodnih zadataka, ali ovoga puta nemamo default blok koda.

|  |
| --- |
| **Napomena**    Nakon izvršavanja zadataka za vežbu preporučuje se i testiranje koda. Kod možete testirati dodeljivanjem novih vrednosti deklarisanim promenljivama u okviru zadataka i na taj način proveriti ponašanje switch–case struktura sa drugačijim uslovnim izrazima. |