Променливи, константи, изрази.

Променливите – това са области от оперативната памет, достъпът до които се осъществява по тяхното име. Всички данни, с които работят програмите се запазват във вид на променливи.

В PHP типът на променливата се установява не от програмиста, а се определя автоматично по време на изпълнение на програмата в зависимост от контекста, в който се използва променливата.

4.1. Имена на променливи

4.1.1. Идентификатори

Имената на променливите са идентификатори, също както имената на функциите и класовете. Идентификаторите може да бъдат с произволна дължина и да съдържат букви, цифри, символ долна черта (_) и символ ($). Те не могат да започват с цифра.

Всички променливи в PHP имат имена, които започват със знак ($), например $my_var. По този начин те лесно се отличават от останалия код.

Имената на променливите са чувствителни към малките и големите букви, т.е. $my_var, $My_var и $MY_VAR са различни променливи. Изключение от това правило са имената на функции – за тях тази разлика не се прави.

Име на променлива може да съвпада с име на функция. Подобна практика трябва да се избягва.

В официалната документация е написано, че името на променливата може да се състои не само от латински букви и цифри, но така също от символи, чиито код е по-голям от 127. Това означава, че променливите могат да съдържат букви от българската азбука. Въпреки това, не се препоръчва използването на българоезични имена на променливите, заради различните кодировки на кирилицата.

Ако е нужно да се укаже тип на променливата, то може да се използва инструкция cast в произволна функция settype().

4.1.2. Присвояване на стойност на променливи

На променлива се присвоява стойност чрез оператора за присвояване (=).

4.2. Типове данни

Типът на променливата сочи типа данни, които тя съхранява.

4.2.1. Тип integer

Това е целочислен тип, обикновено с дължина 32 бита (от -2 147 483 648 до 2 147 483 647). Целите числа може да бъдат задавани в десетична, осмична или шестнадесетична бройна система, с предшестващ знак (+) или (-).

Използването на осмична бройна система се извършва с прибавяне на префикс числото (0) (нула), за използването на шестнадесетична бройна система е нужно пред числото да се постави префикс (0x).

4.2.2. Тип double

Реални числа (числа с двойна точност) може да бъдат определяни с помощта на следващите редове:

4.2.3. Тип string

Тип string се използва за работа с низове от знаци. Низовете са с произволна дължина. Те може да съдържат и нулеви символи.

Еднозначната идентификация на променливите дава възможност за използване на променливите непосредствено в низове. Например:

```
$name = "Иван";

$age = 23;

echo "$name е $age годишен.";
```

След изпълнение на примера се получава низ "Иван е 23 годишен."

Низовете се задават с използването на единични или двойни кавички.

Ако низът е заграден в двойни кавички ("), то променливите участват със своята стойност. Възможно е указването на специални символи (n) – нов ред, (\) - символ (), ()- двойна кавичка, ($) - знак ($).

Вторият способ за задаване на низове е използването на единични кавички ('). Тогава в низа може да се използват само символи (\) и ('). Променливите в такива низове не се обработват.

Обединение на низове се извършва с оператор точка (.).

Един низ може да се оцени числово, ако съдържа символ (.), (e) или (E) – като реално число и в противен случай като цяло число. Значението се определя от началната част на низа. Ако низът започва с допустими числови данни, те се използват в качеството на значение, в противен случай ще бъде оценен като 0.

Допустимите числови данни са незадължителния знак, след който следват една или няколко цифри (може да присъства и десетичната точка), а след тях незадължителнтата експонента. Експонентата, това е (e) или (E), следвано от една или няколко цифри.

```
$my_var = 5 + "12.4";                    //$my_var е реално число (17.4)

$my_var = 1 + "-1.3e3";     //$my_var е реално число (-1299)

$my_var = 7 + "Асен-1.3E3";          //$my_var е цяло число (7)

$my_var = 8 + "Никола";                  //$my_var е цяло число (8)

$my_var = 3 + "12 Димитър";         //$my_var е цяло число (15)

$my_var = "18.1 книги " + 7;//$my_var е реално число (25.1)

$my_var = "6.0" + 9.0;                   //$my_var е реално число (15)

$my_var = "Иван" + "Силвия";//$my_var е цяло число (0)
```

Примерите може да се тестват, допълвайки всеки низ със следния ред:

```
echo "$my_var == $my_var; тип " . gettype($my_var) . "
n";
```

4.2.4. Тип array

Масивът е променлива, която съхранява множество или поредица от стойности. Един масив може да има много елементи. Всеки елемент може да съдържа единична стойност (например текст или число) или друг масив. Масив съдържащ други масиви се нарича многомерен масив.

Определянето на масив се извършва с конструкция array() или непосредствено задавайки стойността на неговите елементи.

```
array ([key0] => value0, [key1] => value1, ...)
```

Езиковата конструкция array() приема като параметри двойки ключ => стойност, разделени със запетаи. Символът (=>) установява съответствието между ключа и стойността. Ключът може да бъде както цяло число, така и низ. Числовите ключове на масива се наричат индекси. Индексирането на масив в PHP започва от нула. Стойностите на елементите от масив може да се получат като след името на масива в квадратни се запише ключа на търсения елемент.

Ако елементът ключ не е зададен, то за ключ се избира максималният числов ключ в масива, увеличен с единица. Ако този максимален елемент е отрицателен, то следващият ключ от масива ще е нула (0).

Използването на TRUE и FALSE като ключове довежда до съответното им преобразуване в единица (1) и нула (0) от тип integer. Като ключ може да се използва NULL – вместо ключ се получава празен низ. Използването на празен низ като ключ е допустимо след записването му в двойни кавички. Като ключове не се използват масиви и обекти.

Задаването на масив със скобите ([) и (]) се извършва като вътре в тях се записва ключа, например $student["MI"]. Указването на нов ключ и стойност, например $student["SA"] = "Методи за транслация", добавя нов елемент в масива. Ако не се укаже ключ, а само се присвои стойност, например, $student[] = "Теория на графите", то новия елемент ще има числов ключ, с единица по-голям от максималния съществуващ. При условие, че масив, в който се добавя елемент, несъществува, то този масив се създава. Промяната на стойността на съществуващ елемент от масив се извършва като по неговия ключ се присвои нова стойност.

Изтриването на елемент от масив се извършва с използване на фунцията unset().

Максималният числов ключ при добавянето на елемент в масив с използване на прази скоби ([]) се търси сред ключовете съществуващи в масива при последното му преиндексиране. Числовото преиндексиране на масив може да се извърши автоматично с използване на функцията array_values().

4.2.5. Тип object

Обектните променливи се използват за съхраняване на инстанции на класове. За достъп до методите на обекта се използва оператор (->). Създаването на нов обект се извършва с израз new.

4.2.6. Тип boolean

Булевият (логически) тип е тип за променливи, съдържащи стойност true (истина) и false (неистина).

4.2.7. Тип NULL

Променливи от тип NULL са тези, на които не е зададена стойност, които са били унищожени с помощта на unset() или им е присвоена стойност NULL.

4.2.8. Тип resource

Стойност от тип ресурс (връзки към външни ресурси) връщат някои вградени функции (например, функциите за работа с бази от данни).

4.2.9. Променливи от тип променлива

4.3. Константи

4.3.1. Дефиниране на константи

Стойностите, съхранявани в променлива, може да се променят. За съхраняване на постоянни величини, чиято стойност не се променя в хода на изпълнение на скрипта, се използват константи. Константата не се унищожава след нейното деклариране. Константите се използват с техните имена без префикс ($). Константите се дефинират с помощта на функция define(), която има следния синтаксис:

define("ИМЕ_НА_КОНСТАНТА",

"стойност_на_константа"

[нечувствителност_към_малки/големи_букви])

По подразбиране константите са чувствителни към малки/големи букви. Това може да се промени чрез задаване на стойност TRUE на аргумента нечувствителност_към_малки/големи_букви. Всички имена на константи се пишат с големи букви, за да се различават от променливите. За получаване стойностите на константите се използва функция constant() с параметър име на константа.

4.3.2. Предопределени константи

Предопределените константи са установени от самия PHP-интерпретатор. Те могат да се прегледат с командата phpinfo(). Командата извежда и допълнителна информация.

4.4. Област на видимост на променливите

Различават се четири области на видимост на променливите:

- Вградените суперглобални променливи са видими навсякъде в PHP-скрипта;

- Декларираните като глобални променливи в скрипта са видими навсякъде в този скрипт, освен в неговите функции;

- Променливи, използвани във функция, са локални за тази функция;

- Декларираните като глобални във функция (с конструкция global) сочат към глобални променливи със същите имена.

Скриптовите суперглобални променливи са следните:

- $GLOBALS – масив от всички глобални променливи;

- $_SERVER – масив от променливи от обкръжението на сървъра;

- $_GET – масив от променливите, подадени на скрипта по метода GET;

- $_POST – масив от променливите, подадени на скрипта по метода POST;

- $_COOKIE – масив от cookie променливи;

- $_FILES - масив от променливи, свързани с качването на файлове (upload);

- $_ENV – масив от променливи от обкръжението;

- $_REQUEST – масив от всички подадени от потребителя променливи (GET, POST и COOKIE);

- $_SESSION – масив от сесийните променливи.