Operatori

- aritmetički operatori (+, - , *, /, %, **)
- operatori dodele (=, +=, -= , *=, /=, %=)
- operatori poređenja (==, ===, !=, <>, !==, >, <, >=,<=, <=>)
- inkrement i dekrement (pre i post tip, ++, --)
- logički operatori (&&, and, ||, or, !, xor)
- string operatori (., .=)
- operatori za rad sa nizovima (+, ==, ===, !=, <>, !==)
- operatori uslovnog dodeljivanja (?:,??) jedina 2 tenarna

NOT

x	x'
0	1
1	0

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

preuzeto sa https://introcs.cs.princeton.edu/java/71boolean/

Kontrolna struktura - Logički izrazi

https://www.php.net/manual/en/language.control-structures.php

- if
- if-else-ifelse
- switch

Fig: If Statement

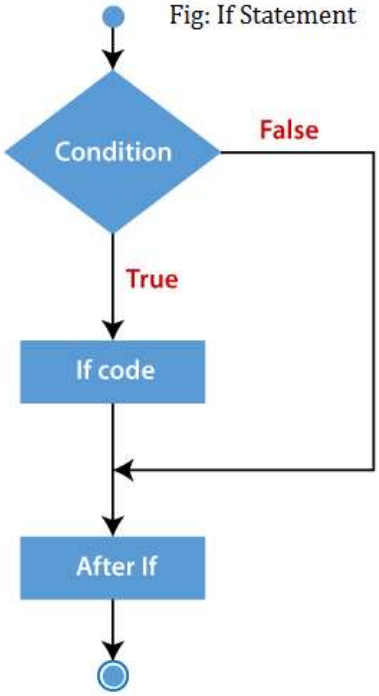


Fig: If-else Statement

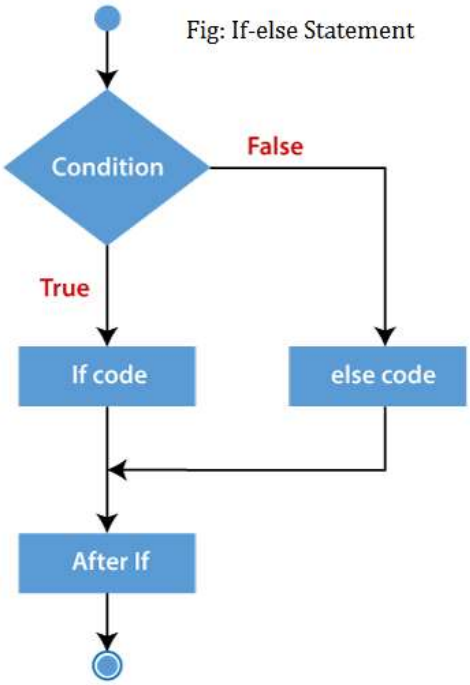
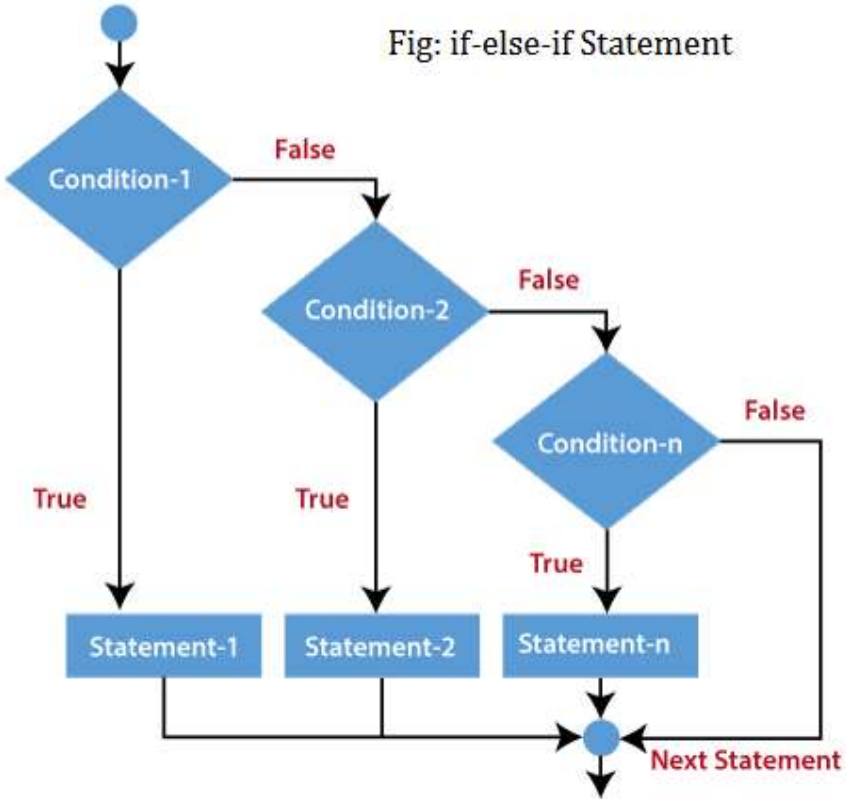
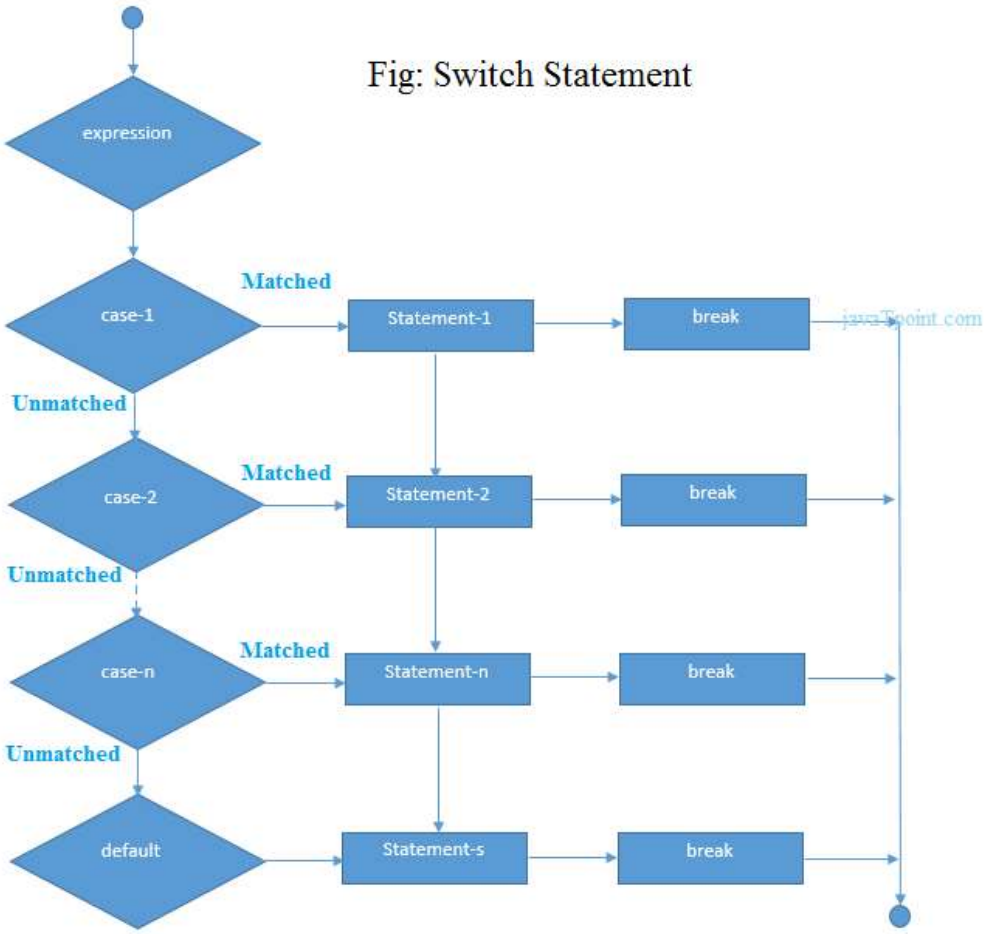


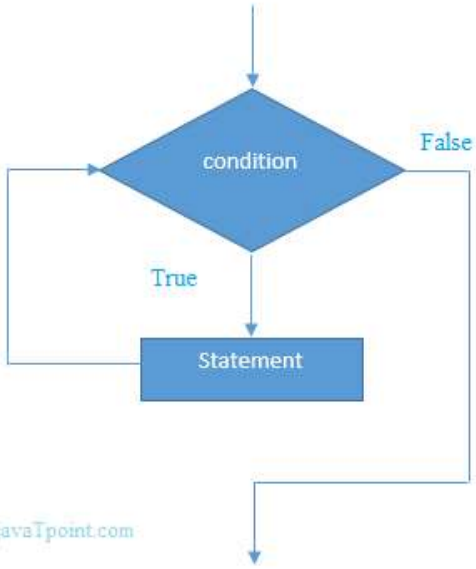
Fig: if-else-if Statement



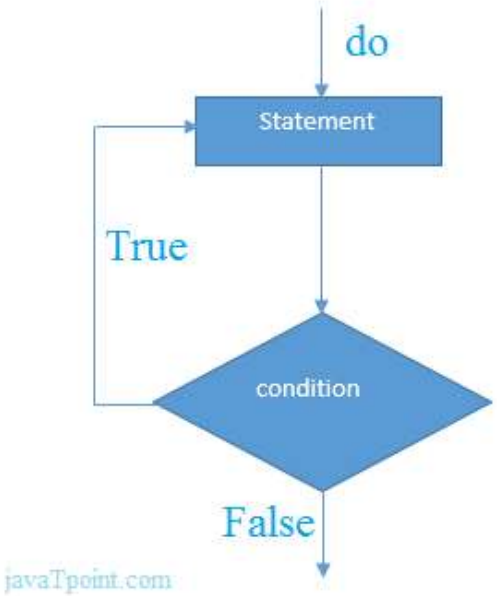


Kontrolna struktura - Petlje

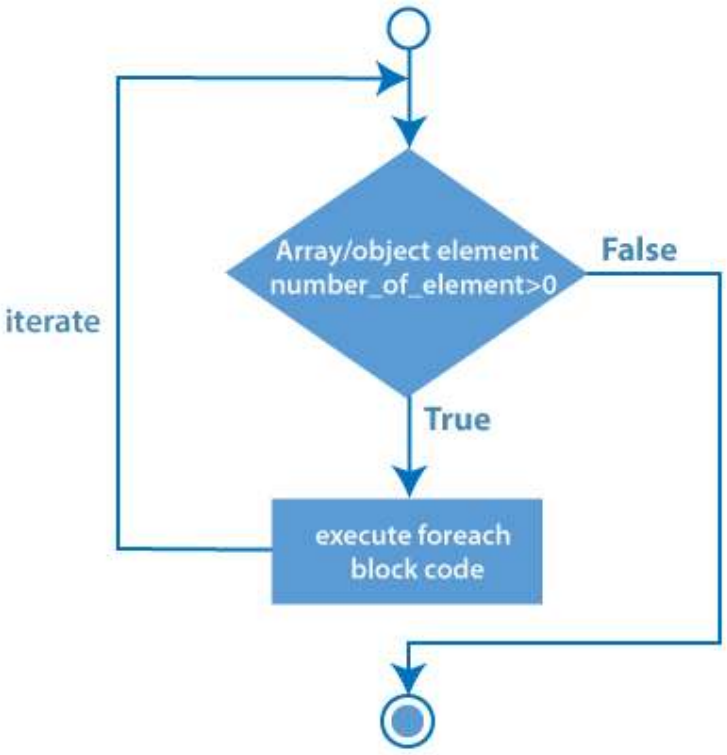
- while
- do-while
- for
- foreach
- break
- continue



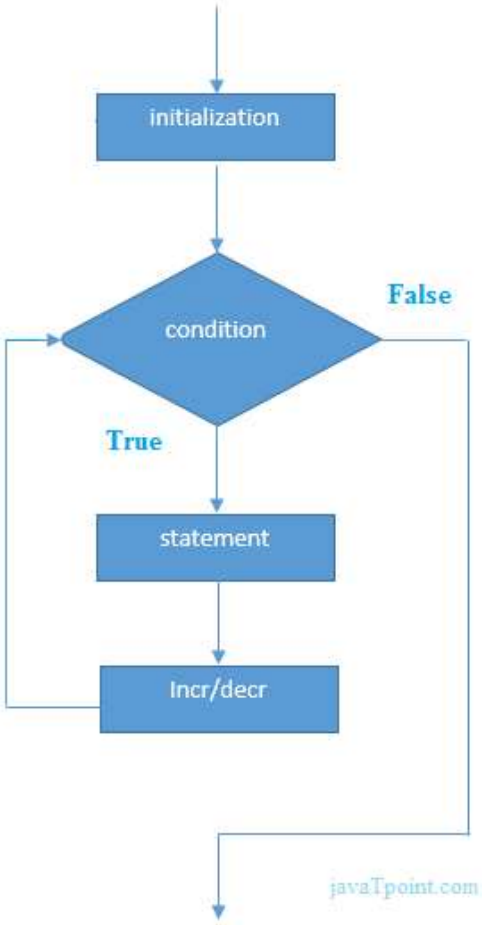
While petlja



do petlja



foreach petlja



for petlja

Svi dijagrami toka su preuzeti sa https://www.javatpoint.com/

Korisničke funkcije

Osim ugrađenih PHP funkcija, moguće je kreirati i sopstvene funkcije.
Funkcija je blok naredbi koji se može više puta koristiti u programu. Neće se izvršiti automatski kada se funkcija definiše, već se mora pozvati.
Funkcija može imati proizvoljan broj argumenata, koji mogu imati predefinisanu vrednost.

```
<?php
//definicija funkcije
function imeFunkcije() {
  echo "Funkcija je pozvana!";
}

 imeFunkcije(); // pozivanje funkcije
```

```
?>
<?php

function pozdravi($ime=”Ivana”) {
  echo $ime . ", zdravo!";
}
//poziv bez argumenta
pozdravi(); // ispis ”Ivana, zdravo!”
//poziv sa argumentom
pozdravi(“Gorane”); // ispis “Gorane, zdravo!”
?>
```

Prikaz grešaka i njihovo otklanjanje

4 vrste grešaka :

- fatalne (fatal) - kada nije moguće izvršiti kod
- sintaksne (syntax) - kada poštoji greška u samom PHP kodu
- upozorenja (warnings) - postoji problem, ali je moguće izvršiti kod
- obaveštenje (notices) - nude savet

Podešavanje prikaza grešaka:

- php.ini fajl

  ```
  display_errors = On
  error_reporting = E_ALL
  ```

- u samom kodu

  ```
  error_reporting(E_ALL);
  ini_set("display_errors", 1);
  ```

Lokacija php.ini fajla /opt/lampp/etc/php.ini ili pogledajte šta kaže funkcija phpinfo()

https://www.php.net/manual/en/errorfunc.constants.php

Ako se kod koji smo napisali izvršava, ali ne na očekivani način, dobar način da saznamo šta se dešava je da koristimo funckiju echo, var_dump() ili debug_print_backtrace();

Spisak funkcija za rad sa greškama - https://www.php.net/manual/en/book.errorfunc.php

Objektno orijentsano programiranje

https://www.php.net/language.oop5

Klasa je tip podatka koji određuje kako će se objekat tog tipa ponašati i šta će objekat sadržavati, odnosno koje karakteristike ga opisuju/određuju.

Stanje objekta opisano je atributima, a ponašanje metodama.

Koje su glavne karakteristike objektno orijentisanog programiranja? (pitanje koje se često postavlja na intervjuima za posao)

- Nasleđivanje - omogućava klasama da naslede svojstva i metode od drugih klasa. Ovo pomaže u poboljšanju ponovnog korišćenja koda.
- Inkapsulacija - omogućava povezivanje podataka i logike zajedno u jednu celinu. Takođe omogućava skrivanje podataka.
- Polimorfizam - sposobnost postojanja u više oblika. Jedan interfejs može se implementirati na više načina pružanjem različitih definicija.
- Apstrakcija podataka - sposobnost objektno orijentisanog programiranja koja omogućava skrivanje detalja implementacije logike, ali omogućava pristup samo važnim informacijama.

Kontrola pristupa vrši se upotrebom tri ključne reči:

- public - podrazumeva se da je promenljiva ili metoda javna ako nisu drugačije definisane. To znači da mu može pristupiti bilo ko i iz bilo kog dela koda.
- private - promenljiva ili metoda definisana kao privatni pristup moguća je samo unutar klase u kojoj se nalazi.
- protected - do koje promenljive ili metode je zaštićen pristup je moguć unutar klase u kojoj se nalazi, ali i unutar nasleđenih klasa

Magične metode su posebne metode koje nadjačavaju podrazumevanu radnju PHP-a kada se na objektu izvode određene radnje. - https://www.php.net/manual/en/language.oop5.magic.php

Apstraktne klase služe samo kao nacrt za klase koje ih nasljeđuju i nije moguće stvoriti objekt apstraktne klase. Korisne su u slučajevima kada želimo izbjeći ponavljanje istih atributa i metoda u više klasa, ili kada želimo osigurati da klase implementiraju određenu metodu. Apstraktne metode mogu biti deklarirane samo unutar apstraktnih klasa i one nemaju implementaciju.

Interfejsi - Nemaju nikakvu implementaciju, samo pobrojane metode koje moraju da imaju klase koje implementiraju taj interfejs..
https://www.php.net/manual/en/language.oop5.interfaces.php

Nasleđivanje - Kada klasa potiče od druge klase. Podređena klasa će naslediti sva javna i zaštićena svojstva i metode od nadređene klase. Osim toga, može imati svoja svojstva i metode. https://www.php.net/manual/en/language.oop5.inheritance.php

Statične metode i funkcije - nemaju $this i pozivaju se ne za objekat nego za celu klasu. Statički atribut je zajednički za celu klasu i postoji samo jedan (za razliku od nestatičkih atributa kojih ima onoliko koliko je objekata kreirano).

Traits - PHP podržava samo jedno nasleđivanje: podređena klasa može naslediti samo jednog roditelja.Traits se koriste za deklarisanje metoda koje se mogu koristiti u više klasa i na taj način se rešava problem dupliranja koda.

Rad sa bazom

SQL je jezik uz pomoć kojeg radimo operacije na bazi. SQL je bazični jezik koji se koristi za sve tipove relacionih baza.
SQL je skraćenica od Structured Query Language i služi za upravljanje RDBMS (Relational database management system) kao što je MySQL.

MySQL je nastao sredinom 90 kao jedan od prvih open-source baza podataka dostupnih na tržištu. Danas postoji više varijanti mysql ali razlike nisu toliko velike jer koriste istu sintaksu i sama funkcionalnost se ne menja mnogo.
Mysql je RDBMS koja čuva podatke u bazi u odredjenoj strukturi.
Najčešća kombinacija MySQL-a je sa PHP i u kombinaciji sa apache web serverom na linux distribucijama. MySQL koristi SQL jezik za upit prema bazi i tabelama.

Razlika je da je SQL jezik kojim radimo operacije unutar baze a MySQL je sama baza tj. engine.

Rad sa bazom je moguć na više načina:

- preko PHPMyAdmin-a
- preko terminala
- kroz kod - glavni naš cilj

https://dev.mysql.com/doc/refman/8.0/en/tutorial.html
https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html

Pristup bazi kroz terminal:
/opt/lampp/bin/mysql -u root
exit

Neke od komandi:

- mysql -u <user> -p<password> <ime_baze> -- komanda za spajanje na bazu
- show databases; -- komanda koja prikazuje sve baze na sistemu
- use <database_name>; -- prebacivanje na odredjenu bazu podataka
- show tables; -- Ispisivanje svih tabela unutar selektovane baze
- describe table_name; -- Describe komanda vraća strukturu polja neke tabele
- show columns from [table name]; -- Slično kao i describe komanda

- drop database [database name]; -- Komanda za drop selektovane baze
- drop table [table name]; -- Komanda za drop neke tabele
- SHOW INDEXES FROM table_name; -- Prikaz svih INDEX polja u nekoj tabeli
- ALTER TABLE new_table_name -- ALTER direktiva koja dodaje polje u tabelu
- ADD
- new_column_name column_definition
- [FIRST | AFTER column_name]
- ALTER TABLE table_name -- ALTER direktiva koja menja definiciju neke kolone/polja
- MODIFY column_name column_definition
- [FIRST | AFTER column_name];
- ALTER TABLE vehicles -- Primer ALTER direktive za menjanje definicije neke kolone/polja
- CHANGE COLUMN note vehicleCondition VARCHAR(100) NOT NULL;
- ALTER TABLE table_name -- Primer ALTER direktive za brisanje polja u tabeli
- DROP COLUMN column_name;
- ALTER TABLE table_name -- Primer ALTER direktive za rename/novo ime neke tabele
- RENAME TO new_table_name;
- ALTER TABLE vehicles
- RENAME TO cars; -- Primer rename tabele

SQL se koristi za pristup, ažuriranju i manipulisanju podataka unutar baze.

Primer baze sa kojom radimo - https://dev.mysql.com/doc/employee/en/

Rad sa bazama u PHPu

U PHP je moguće koristit MySQL bazu koristeći se:

- MySQL ekstenzija (ne koristi se više)
- MySQLi ekstenzija (i je za improved) - proceduralno i objektno
- PDO (PHP Data Objects)

https://www.php.net/manual/en/set.mysqlinfo.php

SQL injection je napad u kojem se maliciozni kod ubacuje u SQL server kako bi izvršio određenu naredbu. Najčešći rezultat ovakvih napada jesu neautorizovani pristupi poverljivim podacima ili uništenje važnih podataka. SQL Injection je jedna od najčešćih metoda napada na webu.

Način da se izborimo sa SQL injection - Prepared Statements and Bound Parameters.

U fajlu za vežbu je primer korišćenja proceduralne MySQLi ekstenzije.