# Лаб: Прости операции и пресмятания

Задачи за упражнение в клас и за домашно към курса ["Основи на програмирането" @ СофтУни](https://softuni.bg/courses/programming-basics).

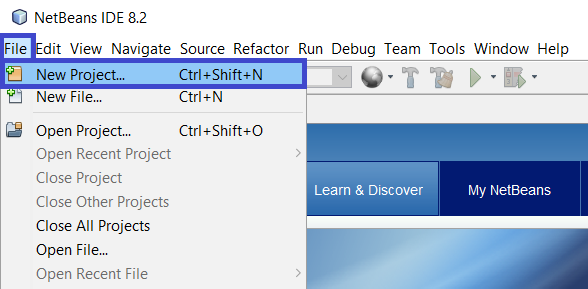
Тествайте решенията си в Judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1011>

## Създайте нов проект NetBeans 8.2

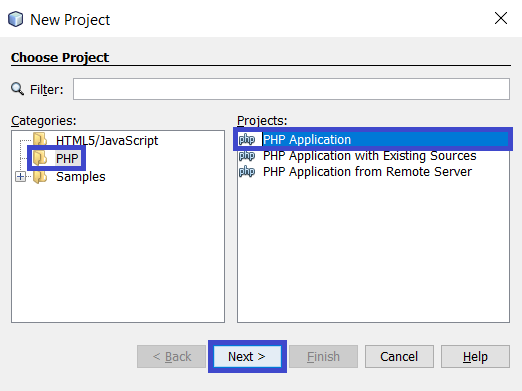
Създайте нов проект (**New Project**) в **NetBeans**. Проектите в NetBeans обединяват **група файлове** (в нашия случай **скриптове**). Тази възможност е изключително удобна, когато искаме да работим по няколко задачи и бързо да превключваме между тях или искаме да обединим логически няколко взаимосвързани файла.

В настоящото практическо занимание ще използваме **нов проект с няколко скрипта** за да организираме решенията на задачите от упражненията – всяка задача в отделен скрипт и всички скриптове в общ проект.

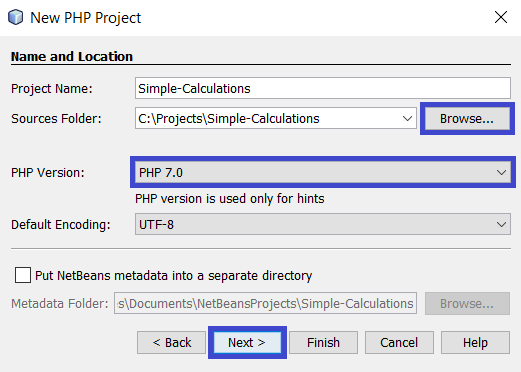
1. Стартирайте **NetBeans**.
2. Създайте **New Project**: [File]🡪 [New Project].



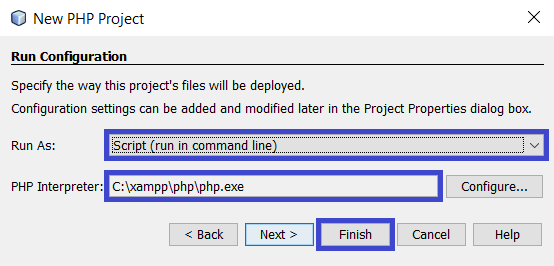
1. Изберете от в прозореца Categories: [**PHP**] 🡪 [**PHP Application**]



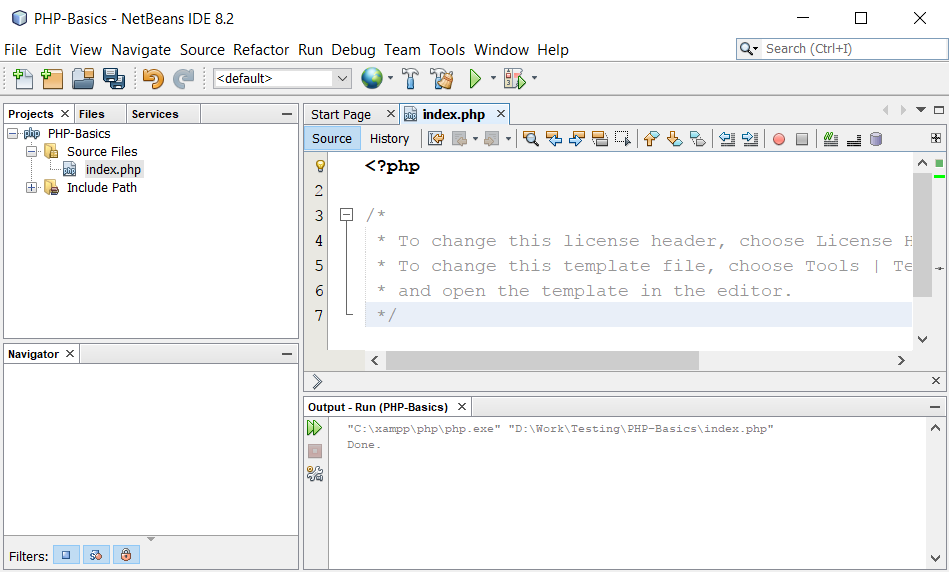
1. Сега може да изберете директория за проекта си, както и версия на **PHP** (ще работим с **PHP 7.0**). След това натиснете **Next** (**не** натискайте **Finish**).



1. Ако сте инсталирали **XAMPP** успешно, **NetBeans** ще намери **PHP интерпретатор** автоматично. Остава само да промените проекта на **Script** (конзолен проект).



1. Това е всичко, натискаме **Finish**.

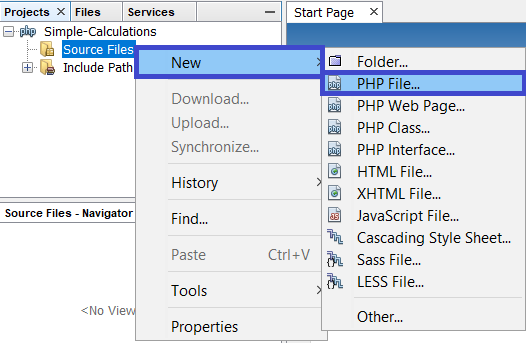


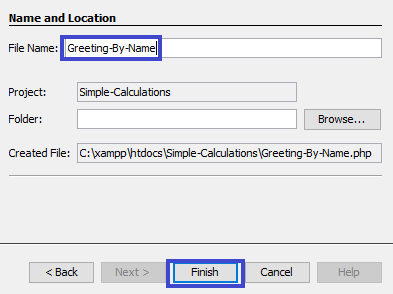
Целта на този проект e да добавяте в него **по един скрипт за всяка задача** от упражненията.

## Поздрав по име

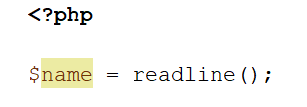
Напишете програма, която **чете от конзолата име на човек** и отпечатва "Hello, <name>!", където <name> е въведеното име от конзолата.

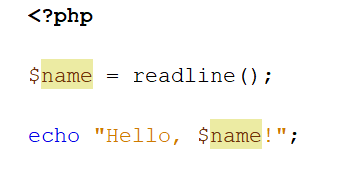
Създайте **нов скрипт (или преименувайте сегашния)** в новосъздадения проект. В проекта си кликнете с десен бутон на мишката върху **Source Files**. Изберете [New] 🡪 [PHP File…]:



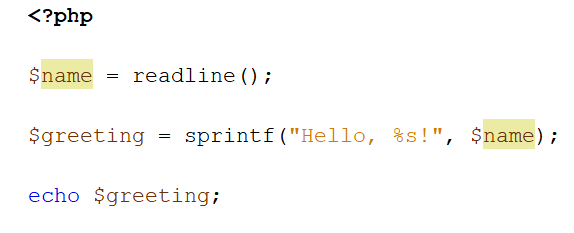


Вече имате проект с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата. Изтрийте генерираният коментар под "<?php" и напишете кода, който е описан в стъпките по – долу.

1. Създайте променливата name и запазете в нея името, което получавате от конзолата  
   
2. Изведете изхода на конзолата като използвате следния шаблон:



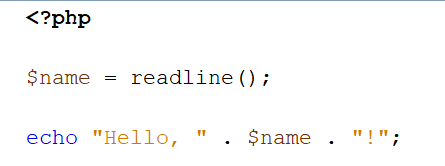
Как работи примера? Методът echo ни позволява да принтираме стойността на променливи на конзолата. Благодарение на двойните кавички, PHP изпълнява кода в тях. Ако кода беше ограден в единични кавички, PHP щеше да интерпретира променливата просто като текст и функцията **echo** нямаше да изпринтира нейната стойност. В нашия случай тази стойност ще бъде стойността на променливата **$**name, която сме прочели от конзолата и сме подали след запетайката в текста за принтиране.



Бихме могли за постигнем същият резултат със **стрингова** **интерполация** (заместване)и функцията **sprintf()**, където стойностите се заместват на мястото на предварително поставени **placeholders.** В примера, който виждаме предварително сме поставили **placeholder** за **стринг** на мястото, където искаме да бъде изпечатана стойността на нашата променлива, а именно **%s** . PHP замества нашият **placeholder** със стойността на желатана променлива **$name**, която подаваме като следващ параметър на функцията **sprintf.**

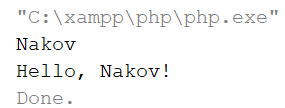
Можете да прочетете повече за **стринговата интерполация** [тук](https://en.wikipedia.org/wiki/String_interpolation), както и за разновидностите от шаблони, които ни предлага езика **PHP**, [тук](http://php.net/manual/en/function.sprintf.php).

Ще дадем и трети последен пример, в който отново можем да постигнем същият ефект и с метода на **конкатенация** (долепяне) на стойности:



В примера долепяме стрингове със стойността на променливи посредством оператора **.** (точка).

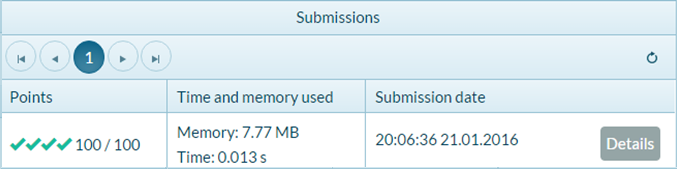
1. **Стартирайте** програмата с [**Shift + F6**] и я **тествайте** с различни входни стойности:



1. **Тествайте** решението в **judge системата (PHP CLI)**

Трябва да получите 100 точки (напълно коректно решение):

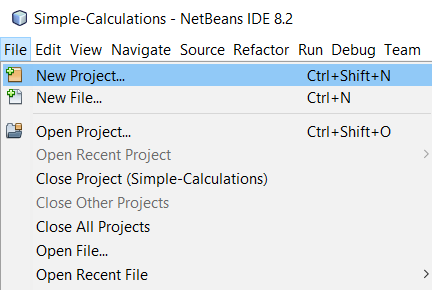


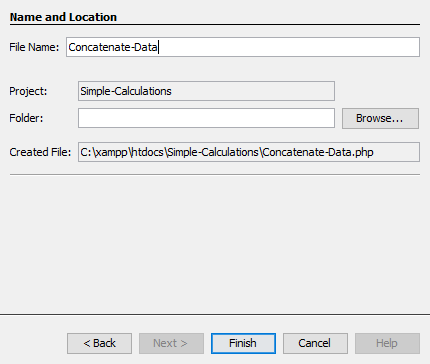


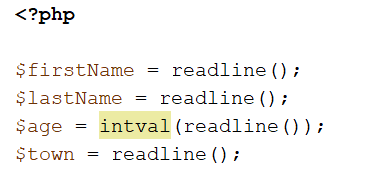
## Съединяване на текст и числа

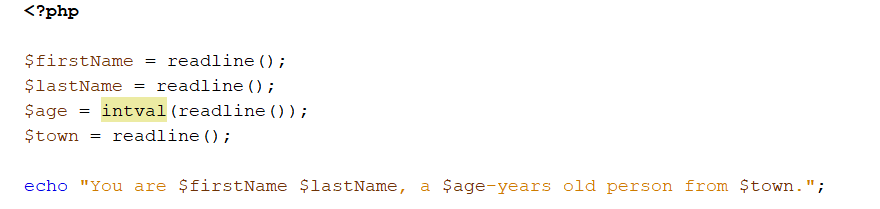
Напишете програма, която прочита от конзолата име, фамилия, възраст и град и печата съобщение от следния вид: "You are <firstName> <lastName>, a <age>-years old person from <town>.

1. Първо създайте **нов PHP конзолен скрипт** в решението “**Simple-Calculations**”. Кликнете с мишката върху решението в File и изберете [File] 🡪 [New File…]:

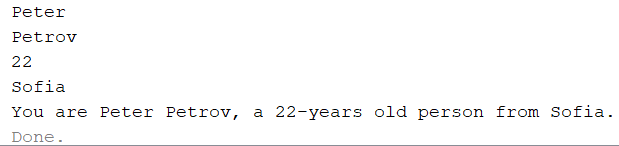




1. Въведете входните данни и ги запишете в променливи с подходящ тип данни:  
    
2. Изведете на конзолата форматирания изход:

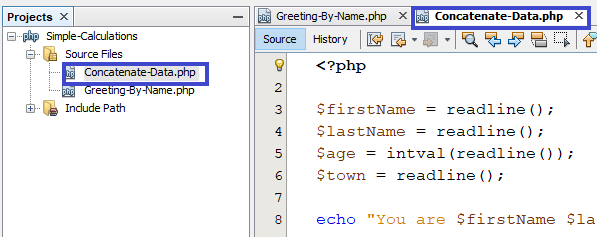


1. **Стартирайте** програмата с [**Shift + F6**] и я **тествайте** с различни входни стойности:



1. **Ако все още получавате резултата от миналата задача? Това се получава, защото не сте сменили скрипта на новия, който желаете да бъде изпълнен. Как да се справите с проблема?**

Сега **превключете от преходната програма** (поздрав по име). Това става с двоен клик на мишката върху файла **Concatenate-Data.php** от предходния скрипт “**Greeting-By-Name**”.



Въведете различни входни стойности и тествайте решението си локално.

1. Тествайтерешението в **judge системата (PHP CLI)**

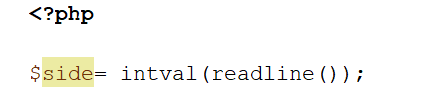
## Пресмятане на лице на квадрат

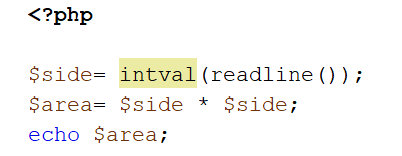
Да се напише **конзолна програма**, която **въвежда цяло число** 'a' и **пресмята лицето на квадрат** **със страна** 'a'.

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| 5 | 25 |

### Насоки

1. **Инициализирайте** променлива side и в нея запишете стойността въведена от конзолата:  
   
2. **Инициализирайте втора променлива** area, в която да запишете стойността за лицето на правоъгълника, получена по формулата side \* side. Принтирайте полученият резултат:

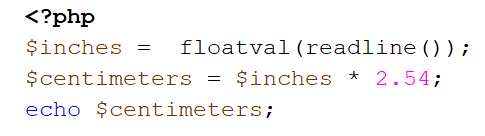


## От инчове към сантиметри

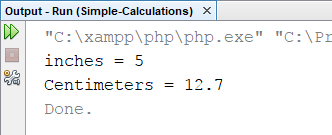
Да се напише програма, която **чете от конзолата число** (не непременно цяло) и преобразува числото **от инчове в сантиметри**. За целта **умножава инчовете по 2.54** (защото 1 инч = 2.54 сантиметра).

### Примерен вход и изход

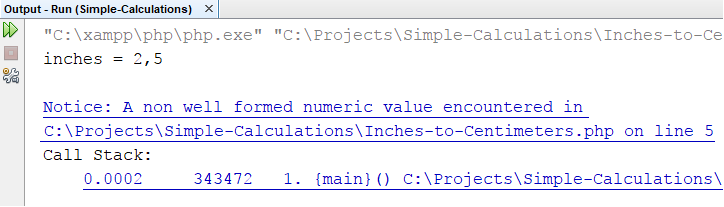
|  |  |
| --- | --- |
| **вход** | **изход** |
| 5 | 12.70 |

1. Инициализирайте променлива с имеinches**,** която приема стойност от конзолата. След това инициализирайте променливата centimeteres, която служи за изчисляване на сантиметрите. Накрая изведете резултата на конзолата, като го **форматирате до втория знак** след десетичната запетая:  


2. **Стартирайте програмата**, както обикновено с [Shift+F6]. Ще се стартира текущата отворена програма, която преобразува инчове в сантиметри.



**Внимание:** в зависимост от регионалните настройки на операционната система, е възможно вместо **десетична точка** (US настройки) да се използва **десетична запетая** (BG настройки). Ако програмата очаква десетична точка и бъде въведено число с десетична запетая или на обратно (бъде въведена десетична точка когато се очаква десетична запетая), ще се получи следната грешка:



Препоръчително е **да промените настройките на компютъра си**, така че да се използва **десетична точка**:





## Лице на трапец

Напишете програма, която чете от конзолата три числа b1, b2 и h и **пресмята лицето на трапец** с основи b1 и b2 и височина h. **Формулата за лице на трапец** е (b1 + b2) \* h / 2.

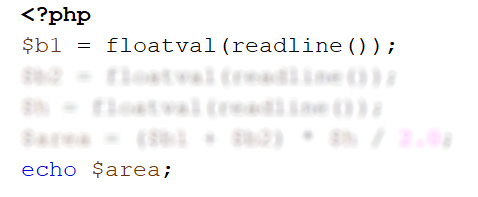
На фигурата по-долу е показан трапец със страни 8 и 13 и височина 7. Той има лице (8 + 13) \* 7 / 2 = 73.5.

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| 8  13  7 | 73.50 |

### Насоки

1. Добавете към текущия проект още един **конзолен PHP скрипт** с име “Trapezoid-Area”.
2. **Напишете кода**, който чете входните данни от конзолата, пресмята лицето на трапеца и го отпечатва:



Кодът на картинката е нарочно размазан, за да си го доизмислите и допишете сами.

1. **Тествайте** решението локално с [Shift+F6] и въвеждане на примерни данни.

## Периметър и лице на кръг

Напишете програма, която чете от конзолата **число** r и пресмята и отпечатва **лицето** и **периметъра на кръг** / **окръжност** с радиус r, като **форматирате изхода до втория знак** след десетичната запетая със **закръгляне**.

### Примерен вход и изход

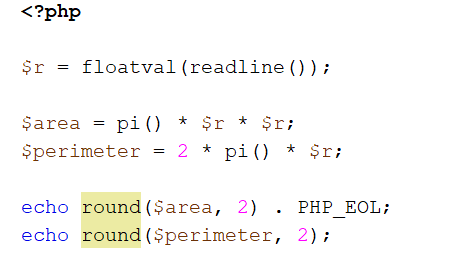
|  |  |
| --- | --- |
| **вход** | **изход** |
| 3 | 28.27  18.85 |
| 4.5 | 63.62  28.27 |

**Насоки**

За изчисленията можете да използвате следните формули:

* area = pi() \* r \* r
* perimeter = 2 \* pi() \* r

1. Създайте променливата r, която записва вход от конзолата, който е реално число.
2. След това създайте променливите area и parameter, които записват изчисленията за площ и периметър.
3. За да изчислите правилно площта и периметъра използвайте вградената функция pi(), която репрезентира числото **π** в програмирането.
4. Закръглете резултата до вторият знак след десетичната запетая посредством фунцията [**round()**](http://php.net/manual/en/function.round.php)



1. Изведете резултата на конзолата в желания формат.

## Лице на триъгълник

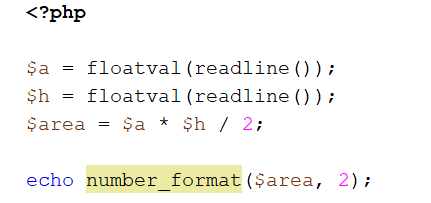
Напишете програма, която чете от конзолата **страна** и **височина** на **триъгълник** и пресмята неговото лице. Използвайте **формулата** за лице на триъгълник: area = a \* h / 2. **Форматирате изхода до втория знак** след десетичната запетая.

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| 20  30 | 300.00 |
| 15  35 | 262.50 |
| 7.75  8.45 | 32.74 |
| 1.23456  4.56789 | 2.82 |

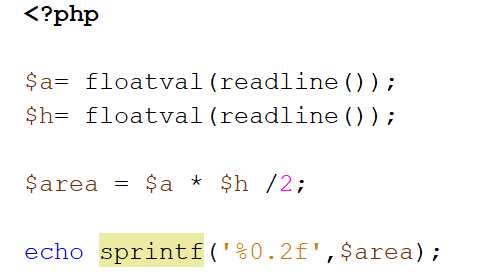
**Насоки**

1. Създайте две променливи с подходящо име и тип, които получават входни данни от конзолата.
2. Създайте трета променлива, която записва резултата от изчислението на лицето.
3. Изведете резултата на конзолата, като използвате примерния изход.



Защо използваме функцията **number\_format** вместо функцията **round**? Причината е в това как работят двете функции. **round** автоматично конвертира стойността получена от функцията в число с плаваща запетая **(float)** и **закръгля числото**. В случай, че полученият резултат е цяло число, **round** ще премахне нулите след десетичната запетая и те няма да бъдат визуализирани. По условие задачата изисква да бъде изпечатан резултата задължително до **два знака** след десетичната запетая. От друга страна функцията **number\_format** превръща резултата във **форматиран** **низ** **(string)** или стрингов еквивалент на числото.

Повече за **number\_format** можете де прочетете [тук](http://php.net/manual/en/function.number-format.php).



Същият резултат ще постигнем ако използваме функцията **sprintf**, която използвахме в първа задача. **'%0.2f'** означава, че искаме резултатът да се форматира в стрингов еквивалент на число с плаваща запетая, форматирано до вторият знак.

## Конзолен конвертор: от градуси °C към градуси °F

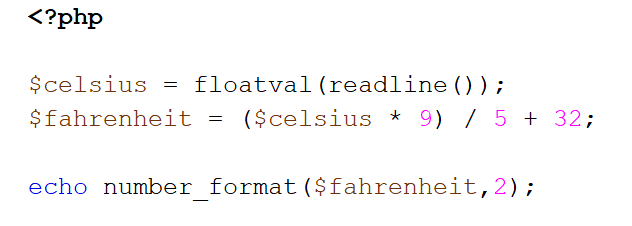
Напишете програма, която чете **градуси по скалата на Целзий** (°C) и ги преобразува до **градуси по скалата на Фаренхайт** (°F). Потърсете в Интернет подходяща [формула](https://www.google.bg/search?q=%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D0%B0+%D1%86%D0%B5%D0%BB%D0%B7%D0%B8%D0%B9+%D1%84%D0%B0%D1%80%D0%B5%D0%BD%D1%85%D0%B0%D0%B9%D1%82&gws_rd=cr&dcr=0&ei=cm7mWaqGJ8ana6i-kcgH), с която да извършите изчисленията. **Форматирате изхода до втория знак** след десетичната запетая.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 25 | 77.00 | 0 | 32.00 | -5.5 | 22.10 | 32.3 | 90.14 |

### Насоки

1. Създайте променлива с подходящо име и тип, която записва вход от конзолата, който е реално число.
2. Създайте втора променлива, която записва конвертирания резултат от градуси °C към градуси °F.



# Примерни изпитни задачи

## \* Рожден ден

За рождения си ден Любомир получил аквариум с формата на паралелепипед. **Първоначално прочитаме от конзолата на отделни редове размерите му – дължина, широчина и височина в сантиметри.** Трябва да се пресметне колко литра вода ще събира аквариума, ако се знае, че определен процент от вместимостта му е заета от пясък, растения, нагревател и помпа.

Един литър вода се равнява на един кубичен дециметър/ 1л=1 дм3/.

**Да се напише програма, която изчислява литрите вода, която са необходими за напълването на аквариума.**

### Вход

От конзолата се четат **4 реда**:

1. **Дължина в см – цяло число в интервала [10 … 500]**
2. **Широчина в см – цяло число в интервала [10 … 300]**
3. **Височина в см – цяло число в интервала [10… 200]**
4. **Процент**  **– реално число в интервала [0.000 … 100.000]**

### Изход

Да се отпечата на конзолата **едно число**:

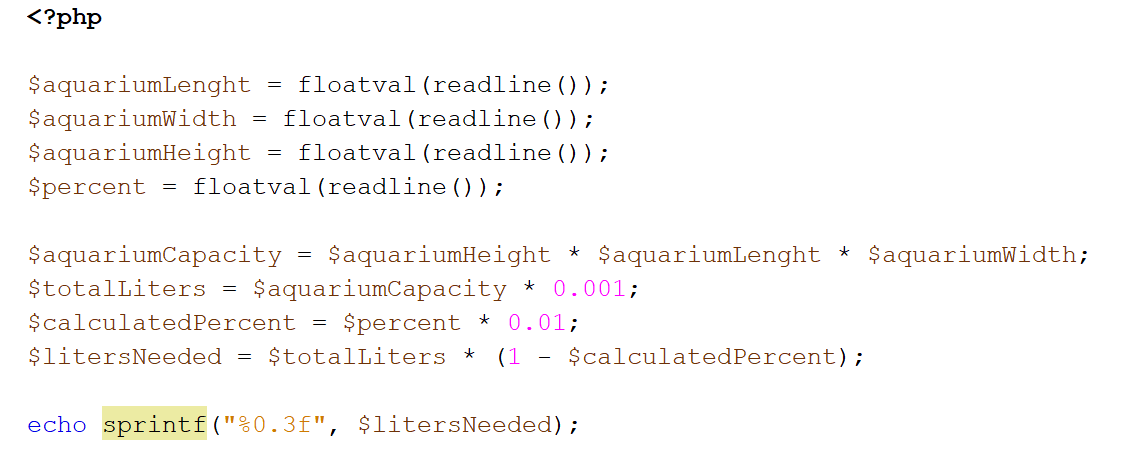
* **литрите вода, които ще събира аквариума**, **форматирани до третия знак след десетичната запетая**.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 85  75  47  17 | 248.689 | Изчисляваме **обем на аквариум**:  **обем на аквариум**= 85\*75\*47=**299625** см3  **общо литри, които ще събере:** 299625 \* 0.001=**299.625** литра  **процент:** 17\*0.01=**0.17**  **литрите, които реално ще трябват :** 299.625\*(1-0.17) = **248.68875 литра** |
| **Вход** | **Изход** |  |
| 105  77  89  18.5 | 586.445 |  |

### Насоки

1. Създайте **четири променливи** с **подходящо име и тип**, които записват вход от конзолата, който е реално число.
2. Създайте **още една променлива**, която записва **резултата за обема на аквариума**.
3. Създайте още една променлива, която записва **резултата за общите литри**, които аквариумът ще събере.
4. Изчислете **процентите**.
5. Накрая създайте **още една променлива**, която записва **литрите**, които реално ще трябват.
6. Изведете резултата на конзолата, като **форматирате** до **третия знак** след десетичната запетая.



За фоматиране на резултата отново използваме **sprint.** Прочетете повече за функцията [тук](http://php.net/manual/en/function.sprintf.php).