**echo**

* Outputs one or more strings separated by commas
* No return value

e.g. echo "String 1", "String 2"

**print**

* Outputs only a single string
* Returns 1, so it can be used in an expression

e.g. print "Hello"

or, if ($expr && print "foo")

**print\_r()**

* Outputs a human-readable representation of any *one* value
* Accepts not just strings but other types including arrays and objects, formatting them to be readable
* Useful when debugging
* May return its output as a return value (instead of echoing) if the second optional argument is given

**var\_dump()**

* Outputs a human-readable representation of *one or more* values separated by commas
* Accepts not just strings but other types including arrays and objects, formatting them to be readable
* Uses a different output format to print\_r(), for example it also prints the *type* of values
* Useful when debugging
* No return value

**var\_export()**

* Outputs a human-readable *and PHP-executable* representation of any *one* value
* Accepts not just strings but other types including arrays and objects, formatting them to be readable
* Uses a different output format to both print\_r() and var\_dump() - resulting output is valid PHP code!
* Useful when debugging
* May return its output as a return value (instead of echoing) if the second optional argument is given

Notes:

* Even though print can be used in an expression, I recommend people avoid doing so, because it is bad for code readability (and because it's unlikely to ever be useful). The precedence rules when it interacts with other operators can also be confusing. Because of this, I personally don't ever have a reason to use it over echo.
* Whereas echo and print are language constructs, print\_r() and var\_dump()/var\_export() are regular functions. You don't need parentheses to enclose the arguments to echo or print (and if you do use them, they'll be treated as they would in an expression).
* While var\_export() returns valid PHP code allowing values to be read back later, relying on this for production code may make it easier to introduce security vulnerabilities due to the need to use eval(). It would be better to use a format like JSON instead to store and read back values. The speed will be comparable.