

Basics

Set	foo = "bar";
Variable	
Define	arrBeer = ["Guinness", "Harp",
Array	"Boddingtons"]; //CF10
Define	structBeer = {isGood:"heck
Struct	yes"}; //CF10
CFAbort	abort;
CFDump	writeDump(beer);
CFInclude	include "template.cfm";
CFOutput	writeOutput("I like " & beer);
CFParam	param name="variables.pint"
	default="1"; //CF9
CFSetting	setting enablecfoutputonly="tr-
	ue" requesttimeout="180"
	showdebugoutput="no"; //CF10

Comments

//write single line comments like this
/ multi-line comments*
are done like this/*

single line comments can follow code:
 var a = 1; *// set up array counter*

/ code can follow multi-line comments:*
like this/* var beer = "tasty";

If / Else If / Else

```
if (beer EQ "Guinness") {
  writeOutput("It's Stout");
} else if (beer IS "Harp") {
  writeOutput("It's Lager");
} else {
  writeOutput("It might be beer...");
}
```

Switch / Case

```
switch(beer) {
  case "Guinness":
    writeOutput("It's Stout");
    break;
  case "Harp": case "Stella":
    writeOutput("It's a Lager");
    break;
  default:
    writeOutput("It's beer, I think...");
}
```

Try / Catch / Throw / Finally / Rethrow

```
try {
  throw(message="Oops", detail="xyz");
  //CF9
} catch (any e) {
  writeOutput("Error: " & e.message);
  rethrow; //CF9
} finally { //CF9
  writeOutput("I always run - even if no
  error is thrown");
}
```

Call Component Function (CF10)

```
result = invoke(
  "beermenu.cfc.beer", // component
  "getRatings", // function
  {beer_id=id} // arguments
);
```

the return of the function is stored in #result#

FOR Loops

```
for (i=1;i LTE arrayLen(array);i+i+1) {
  writeOutput(array[i]);
}
```

FOR IN Loops

```
struct = {};
struct.one = "1";
struct.two = "2";
for (key in struct) {
  writeOutput(key);
}
```

OUTPUT: onetwo

WHILE Loop

```
x = 0;
while (x LT 5) {
  x = x + 1;
  writeOutput(x);
}
```

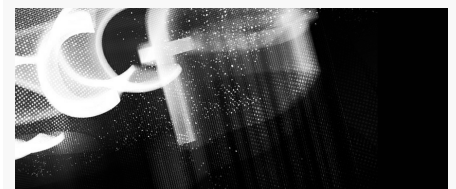
OUTPUT: 12345

DO WHILE Loop

```
x = 0;
do {
  x = x+1;
  writeOutput(x);
} while (x LTE 0);
```

OUTPUT: 1

Adobe ColdFusion



Database Queries (CF9)

```
// create a query object and parameters
qry = new query(
  name="qImports",
  datasource=variables.dsn,
  username=variables.user,
```



By **CChain (Veloz)**
cheatography.com/veloz/

Published 31st July, 2013.
 Last updated 16th March, 2023.
 Page 1 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Database Queries (CF9) (cont)

```
password=variables.password
);
qry.addParam(
    name="beer_id",
    cfsqltype="cf_sql_varchar",
    value=variables.beer_id
);
doQry = qry.execute(sql="
    SELECT b.name,
           m.name as brewer,
           b.alcohol,
           b.IBU,
           b.ratings
    FROM beers as b
    JOIN brewers m ON b.brewer = m.id
    WHERE id = :beer_id and import = true
");
// put the results into the qBeer variable
qBeer = doQry.getResult();
```

Array Iteration (CF10)

```
arrayEach(arrBeer, function(i) {
    writeOutput("I like " & i & "<br/>");
});
```

Array Filtering (CF10)

```
delicious = arrayFilter(arrBeer, function(b) {
    return b.alcohol >= 5 || b.brand EQ
    "Franziskaner";
});
```

Defining Components

```
component extends="Beer" output="false" {
    property name="variety" type="string";
    public boolean function isGood() {
        return true;
    }
    private void drink(required numeric pint) {
        //do stuff
    }
}
```

Defining Functions

```
remote string function drink(required
numeric pints) {
    var dry = 0;
    if (arguments.pints GT dry) {
        return "beer";
    } else {
        return "no beer";
    }
}
```

Transactions

```
transaction {
    //do stuff
    if (good) {
        transaction action="commit";
    } else {
        transaction action="rollback";
    }
}
```

Database Stored Procedures (CF9)

```
// instantiate the SP
sp = new storedproc(
    datasource=variables.dsn,
    procedure="getBeerRating",
    username=request.dsnuser,
    password=request.dsnpassword
);
// set the input var
sp.addParam(
    type="in",
    cfsqltype="cf_sql_integer",
    value=id
);
// set the output var
sp.addParam(
    type="out",
    cfsqltype="cf_sql_bit",
    variable="rating"
);
// execute the SP call
result = sp.execute();
// store the SP return
rating = result.getProcOutVariables().rating;
```

