

Source (snippets/java/StaticInit.java)

```
class StaticInit {
    public static void main(String[] args) {
        if (Mathf.pi-3.1415<0.0001)
            System.out.println("Hello world");
        else
            System.out.println("Hello strange universe");
    }
}
```

Source (snippets/java/Mathf.java)

```
class Mathf {
    static double pi=3.1415;
    static { System.out.println("Bad things happen!");
            System.exit(0); }
}
```

Source (snippets/js/cast.js)

```
document.write("'0'", '0'==0?"==">"<","0 and ");
document.write("0", 0=='0.0'?"==">"<","'0.0' and ");
document.write("'0'", '0'=='0.0'?"==">"<","'0.0'<br />");

document.write(1+2+'X'); document.write('<br />');
document.write('X'+1+2); document.write('<br />');
document.write('X'+(1+2)); document.write('<br />');
```

[C] Faire mauvaise impression 1/2

Source (snippets/c/stringformat.c)

```
#include <stdio.h>

int sfa() {
    char *f="%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x";
    printf(f); printf("\n"); return 0;
}

int main(void) {
    int secret=0x40414243;
    sfa();
    return 0;
}
```

[C] Faire mauvaise impression 2/2

Source (snippets/c/stringformat2.c)

```
#include <stdio.h>

int main(void) {
    char *f="%x.%x.%x.%x.%x.%x.%n";
    int s=0x40414243;
    int *p=&s;
    printf(f); printf("\n");
    if (s==0x40414243) printf("Hello world\n");
    else printf("Bad things happen! Secret is %08x\n",s);
    return 0;
}
```

Source (snippets/ocaml/mutable.ml)

```
let alert test =  
    if test then "Tout va bien" else "Tout va mal!";;  
  
alert true;;  
alert false;;  
  
(alert false).[8]<- 'b'; (alert false).[9]<- 'i';  
(alert false).[10]<- 'e'; (alert false).[11]<- 'n';;  
  
alert false;;
```

Source (snippets/ocaml/mutableexc.ml)

```
let alert test =  
  if test then failwith "minor" else failwith "major";;  
  
let reaction test =  
  try ignore (alert test)  
  with Failure "minor" -> ()  
  | Failure "major" -> failwith "major";;  
  
try alert false with Failure x -> (x.[1]<- 'i'; x.[2]<- 'n');;  
  
reaction false;;
```

Source (snippets/ocaml/hsm.ml)

```
module type Crypto = sig val id:int end;;

module C : Crypto =
struct
  let id=Random.self_init(); Random.int 8192
  let key=(Random.self_init(); let s=Random.int 8192 in
    Printf.printf "< id=%d, key=%d >\n" id s; s)
end;;
```

Source (snippets/ocaml/hsmoracle.ml)

```
let rec oracle o1 o2 =
  let o = (o1 + o2)/2 in
  let module O = struct let id=C.id let key=o end in
  if (module O:Crypto)>(module C:Crypto)
  then oracle o1 o
  else (if (module O:Crypto)<(module C:Crypto)
         then oracle o o2
         else o);;

oracle 0 8192;;
```