1. Capitalizing Constructors

var adam = new **Person**();

2. Separating Words

camel case - type the words in lowercase, only capitalizing the first letter in each word.

upper camel case, as in  MyConstructor(),

lower  camel  case,  as  in  myFunction(), calculateArea()and getFirstName()

variable names - first\_name,  favorite\_bands,  and old\_company\_name.

ECMAScript uses camel case for both methods and properties, although the multiword property  names are rare (lastIndex and  ignoreCase properties of regular expression objects).

3. Other Naming Patterns

Constants - Number.MAX\_VALUE

// precious constants, please don't touch

var PI = 3.14,

MAX\_WIDTH = 800;

Naming globals with all caps can reinforce the practice of minimizing their number and can make them easily distinguishable.

use an underscore prefix to denote a private method or property.

[复制代码](javascript:void(0);)

var person = {

getName: function () {

return this.\_getFirst() + ' ' + this.\_getLast();

},

\_getFirst: function () {

// ...

},

\_getLast: function () {

// ...

}

};

[复制代码](javascript:void(0);)

**Note** that JSLint will complain about the underscore prefixes, unless you set the option nomen: false.

Following are some varieties to the \_private convention:

• Using a trailing underscore to mean private, as in name\_ and getElements\_()

• Using  one  underscore  prefix  for  \_protected properties  and  two  for  \_\_private properties

• In Firefox some internal properties not technically part of the language are available, and they are named with a two underscores prefix and a two underscore suffix, such as \_\_proto\_\_ and \_\_parent\_\_