

## Yahoo User Interface



YUI 2.4.0

# Index

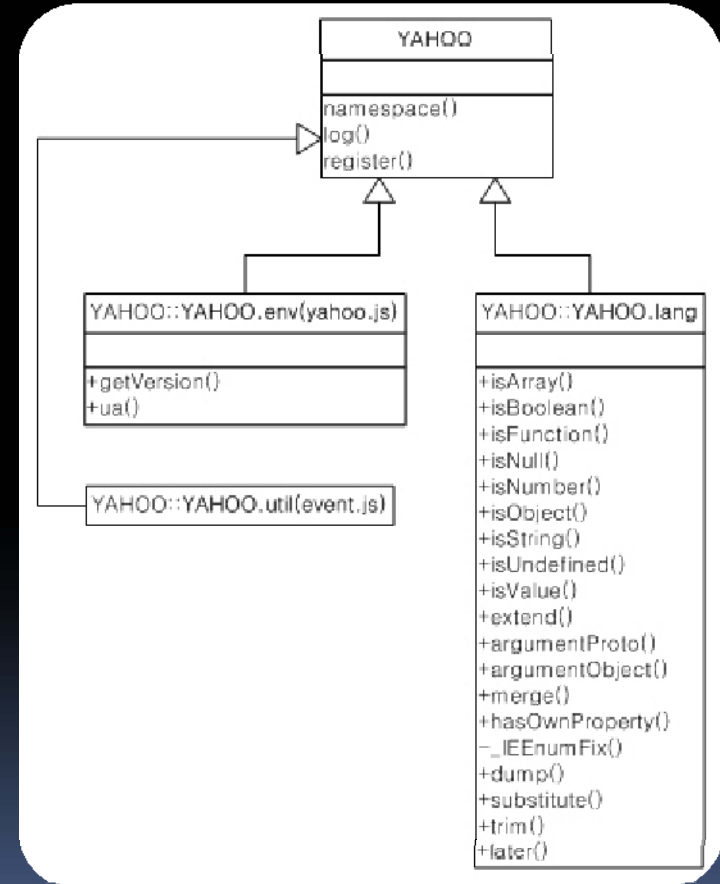
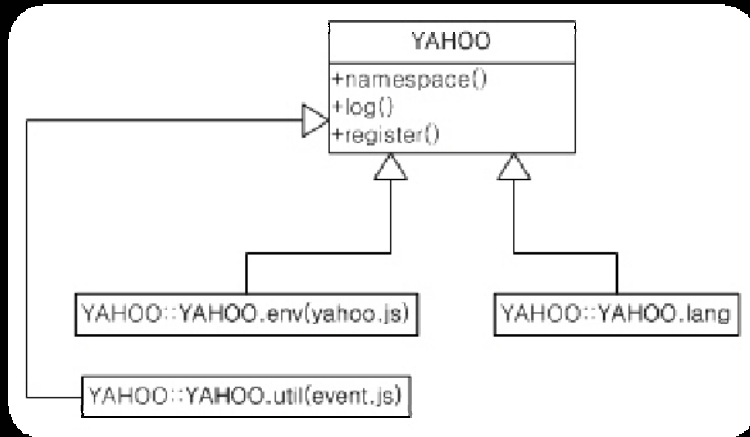
- Introduce Yahoo Global Object
- Object Outline
- Description
- Usage

# Introduce

- **YAHOO global object**는 어떤 구현상 처리를 요구하지 않습니다.
- **YAHOO global object**는 YUI 라이브러리를 구성하는 모든 컴포넌트들의 유용한 메서드들의 프로바이더와 컨테이너 역할을 합니다.

<b>YAHOO Object: Default Members</b>		
<small>See online docs for complete documentation on each default member of the YAHOO object.</small>		
Member	Type	Description
env	object	Environment object. Contains information about what YUI modules are loaded and provides a method for obtaining version information.
example	object	An empty object used as a namespace for example implementations.
lang	object	Contains utility methods. Full list at right.
util	object	Namespace for YUI utilities. Do not add your own members to this object
tool	object	Namespace for developer tools like YUITest. Do not add your own members to this object.
widget	object	Namespace for YUI controls (widgets). Do not add your own members to this object.
log	method	Calls <code>YAHOO.widget.Logger.log</code> ; prevents log messages from throwing errors when the Logger Control is not present.
register	method	Registers a module with the YAHOO object.

# Object Outline



# Type checking

method	Static Method YAHOO.lang.isArray();
arguments	Any
return	Boolean
description	Argument 의 type이 Array 인지 아닌지 체크
usage	<pre>YAHOO.lang.isArray([1, 2, 3]); //-&gt; true YAHOO.lang.isArray(true); //-&gt; false YAHOO.lang.isArray(3384); //-&gt; false YAHOO.lang.isArray(`rhio.kim`); //-&gt; false YAHOO.lang.isArray(new Array()); //-&gt; ??? YAHOO.lang.isArray([]); //-&gt; ??? YAHOO.lang.isArray({}); var a = YAHOO.lang.isArray;     a({}); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isBoolean();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Boolean 값 인지 아닌지 체크
usage	<pre>YAHOO.lang.isBoolean(true); //-&gt; true YAHOO.lang.isBoolean(false); //-&gt; ??? YAHOO.lang.isBoolean(1234); YAHOO.lang.isBoolean([]); YAHOO.lang.isBoolean('ajax professional group'); YAHOO.lang.isBoolean(undefined); YAHOO.lang.isBoolean(null); //-&gt;false YAHOO.lang.isBoolean(!undefined); //-&gt; ??? YAHOO.lang.isBoolean(!null); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isFunction();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Function 인지 아닌지 체크
usage	<pre>function Foo() { ... } YAHOO.lang.isFunction(function Foo() { ... }); //-&gt; ??? YAHOO.lang.isFunction(new Foo()); //-&gt; ??? YAHOO.lang.isFunction(new Object()); //-&gt; ??? YAHOO.lang.isFunction(new Function()); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isNull();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Null 인지 아닌지 체크
usage	<pre>YAHOO.lang.isNull(null); //-&gt; true YAHOO.lang.isNull(undefined); //-&gt; false YAHOO.lang.isNull(false); //-&gt; false YAHOO.lang.isNull(); //-&gt; ??? function a() { return } YAHOO.lang.isNull(a()); //-&gt; ???</pre>



# Type checking

method	Static Method YAHOO.lang.isNumber();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Number형 인지 아닌지 체크
usage	<pre>YAHOO.lang.isNumber(231); //-&gt; true YAHOO.lang.isNumber ('231'); //-&gt; false YAHOO.lang.isNumber ([1,2,3,4].length); //-&gt; true YAHOO.lang.isNumber ('가나다라마바'.length); //-&gt; true YAHOO.lang.isNumber(Number('123123')); //-&gt; true parseInt(4912312312312397862, 16); //-&gt; 2.1566770958534074e+22 YAHOO.lang.isNumber(2.1566770958534074e+22); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isObject();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Object 인지 아닌지 체크
usage	<pre>YAHOO.lang.isObject({}); //-&gt; true YAHOO.lang.isObject(new Object()); //-&gt; true YAHOO.lang.isObject(document.body); //-&gt; true YAHOO.lang.isObject(Function); //-&gt; ??? YAHOO.lang.isObject(Math.round()); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isString();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 String형 인지 아닌지 체크
usage	<pre>YAHOO.lang.isString(''); //-&gt; ??? YAHOO.lang.isString(1234+'alc'); //-&gt; ??? YAHOO.lang.isString(''); //-&gt; ??? YAHOO.lang.isString(`123`*2); //-&gt; ??? YAHOO.lang.isString(`123`*2); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isUndefined();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 Undefined 인지 아닌지 체크
usage	<pre>YAHOO.lang.isUndefined([1, 2, 3]); //-&gt; true YAHOO.lang.isUndefined(true); //-&gt; false YAHOO.lang.isUndefined(3384); //-&gt; false YAHOO.lang.isUndefined('rhio.kim'); //-&gt; false YAHOO.lang.isUndefined(new Array()); //-&gt; ??? YAHOO.lang.isUndefined([]); //-&gt; ??? YAHOO.lang.isUndefined({}); var a = YAHOO.lang.isUndefined;     a({}); //-&gt; ???</pre>

# Type checking

method	Static Method YAHOO.lang.isValue();
arguments	any
return	<i>boolean</i>
description	argument 의 type이 null/undefined/NAN 의 경우 false를 그 외의 것들은 true를 반환 (o, false, 공백)도 true를 반환
usage	<pre>YAHOO.lang.isValue(); //-&gt; false YAHOO.lang.isValue(null); //-&gt; false YAHOO.lang.isValue(undefined); //-&gt; false YAHOO.lang.isValue(NaN); //-&gt; false YAHOO.lang.isValue(o); //-&gt; ??? YAHOO.lang.isValue(""); //-&gt; ??? YAHOO.lang.isValue(false); //-&gt; ???</pre>

# User Agent Detection

- YAHOO.env.ua 오브젝트는 YAHOO global object에 포함되어 있으며 다양한 브라우저(IE, FF, Opera, Safari, Mobile – Opera Mini, iPhone/iPod touch, Nokia N-series)를 구분하는 도구로 쓰입니다.
- 많은 브라우저 스니핑(sniffing) 기술이 있지만 모두 100% 완벽하지는 않습니다. 하지만 크로스 브라우징 대응을 위해서 필요한 부분입니다.

# User Agent Detection

method	Static Method YAHOO.env.ua();
arguments	ua();
return	object
description	YAHOO.env.ua.ie 는 ie 버전 정보를 반환 YAHOO.env.ua.gecko 는 FireFox 버전 정보를 반환 YAHOO.env.ua.opera 는 Opera 버전 정보를 반환 YAHOO.env.ua.webkit 은 Apple Webkit 버전 정보를 반환 YAHOO.env.ua.mobile 은 제한적 Webkit기반 Opera Mini 만 지원하며 버전 정보를 반환
usage	YAHOO.env.ua.ie //-> 0 or 6, YAHOO.env.ua.gecko //-> 0 or 1.8 YAHOO.env.ua.opera //-> 0 or 9.2 YAHOO.env.ua.webkit //-> 0 or 418

# Creating Class Hierarchies

- 자바스크립트는 계층적 클래스 개념(Prototypical Hierarchies)을 제공합니다.
- 다른 객체지향(OOP) 언어들처럼 슈퍼클래스(super class)의 기능적 오버라이드(overrid) 및 확장(extend) 가능한 계층 클래스(class)를 만들 수 있도록 지원합니다.
- `YAHOO.lang.extend()`는 이런 기능을 위한 Static method 입니다.



# YAHOO.lang.extend()

method	Static Method YAHOO.lang.extend();
arguments	extend( <i>object</i> subc, <i>object</i> superc [, <i>object</i> overrides])
return	object
description	superc 가 subc 에 override 됩니다.
usage	<pre>YAHOO.lang.extend(tico, car[, {   getName: function() {},   getBirth: functon() {},   getCompany: function() {} }]);</pre>

# YAHOO.lang.extend()

- extend: `function(subc, superc, overrides) {`
  - `var F = function() {};`
  - `F.prototype = superc.prototype;`
  - `subc.prototype = new F();`
  - `subc.prototype.constructor = subc;`
  - `subc.superclass = superc.prototype;`
  - `if (superc.prototype.constructor == Object.prototype.constructor) {`
  - `superc.prototype.constructor = superc;`
  - `}`
  - `if (overrides) {`
  - `for (var i in overrides) {`
  - `subc.prototype[i] = overrides[i];`
  - `}`
  - `YAHOO.lang._IEEnumFix(subc.prototype, overrides);`
  - `}`
  - `}`

# extend() example

- `var YAHOO.namespace('example');`
- `var Ye = YAHOO.example;`
- `Ye.Bird = function (name) {`
  - `this.name = name;``};`

```
function car(make, model, year)
{ this.make = make, this.model = model, this.year = year }

function hireCar(carNo, make, model, year)
{ this.carNo = carNo, car.call(this, make, model, year) }
```

```
Ye.Bird.prototype.flighted = true; // Default for all Birds
Ye.Bird.prototype.isFlighted = function () { return this.flighted };
Ye.Bird.prototype.getName = function () { return this.name };
```

```
Ye.Chicken = function (name) {
  // Chain the constructors
  this.constructor.superclass.constructor.call(this, name);
};
```

```
// Chickens are birds
YAHOO.lang.extend(Ye.Chicken, Ye.Bird);
```

```
// Define the Chicken prototype methods/members
Ye.Chicken.prototype.flighted = false; // Override default for all Chickens
```

# YAHOO.lang.\_IEEnumFix()

- `_IEEnumFix: function(r, s) {`
- `if (YAHOO.env.ua.ie) {`
- `var add=["toString", "valueOf"], i;`
- `for (i=0;i<add.length;i=i+1) {`
- `var fname=add[i],f=s[fname];`
- `if (YAHOO.lang.isFunction(f) && f!=Object.prototype[fname]) {`
- `r[fname]=f;`
- `}`
- `}`
- `}`
- `}`

# Add Behavior to Objects or Static Classes

- `YAHOO.lang.argumentObject()` 는 `YAHOO.lang.argumentProto()` 와 매우 유사한 메서드입니다.
- 내부적으로 `argumentProto()`는 `argumentObject()`를 사용합니다.
- 클래스 정의(즉 prototype)에 기능적 추가 보다는 오히려 `argumentObject`는 오브젝트 literals과 클래스 인스턴스를 포함한 어떤 오브젝트와도 동작할 수 있게 하기 위함입니다.

# YAHOO.lang.argumentObject()

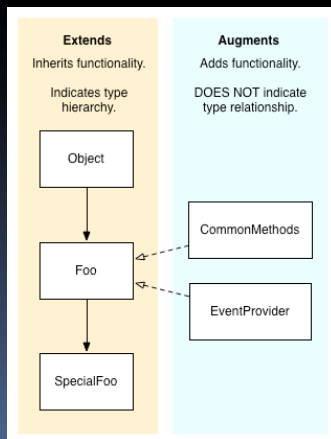
method	Private Method YAHOO.lang.argumentObject();
arguments	argumentObject( <i>object</i> r, <i>object</i> s[, <i>boolean</i>    <i>object</i> override])
return	object
description	Override 가 true 일 경우에 오버라이팅(overwriting) 한다.
usage	YAHOO.lang.argumentObject(widget, dom, event, ui);

# YAHOO.lang.argumentObject()

- augmentObject: `function(r, s) {`
- `if (!s||!r) {`
- `throw new Error("Absorb failed, verify dependencies.");`
- `}`
- `var a=arguments, i, p, override=a[2];`
- `if (override && override!==true) { // only absorb the specified properties`
- `for (i=2; i<a.length; i=i+1) {`
- `r[a[i]] = s[a[i]];`
- `}`
- `} else { // take everything, overwriting only if the third parameter is true`
- `for (p in s) {`
- `if (override || !r[p]) {`
- `r[p] = s[p];`
- `}`
- `}`
- `YAHOO.lang._IEEnumFix(r, s);`
- `}`
- `}`

# Creating a Composition-Based Class Structure

- 컴포지션 기반 클래스 구조 생성을 위한 Static method 입니다.
- YAHOO.lang.argument 와 매우 유사한 메서드입니다.
- Prototype.js 의 Object.extend()와도 유사한 기능을 합니다.
- Foo(superclass)에 새로운 객체 혹은 계층 관계의 새로운 요소를 쉽게 추가할 수 있습니다.
- Foo는 CommonMethods 에 의해 작동하고 EventProvider 에 의해서 사용자 이벤트를 발생시킬 수 있습니다.



## Composite 패턴

1. 부분-집합 관계의 계층도를 표현해야 할 때
2. 복합 객체와 단일 객체 간의 사용 방법에 차이를 두고 싶지 않을 때
3. 계층관계의 요소를 쉽게 추가할 필요가 있을 때



# YAHOO.lang.argumentProto()

method	Static Method YAHOO.lang.argumentProto();
arguments	argumentProto( <i>object r</i> , <i>object s</i> )
return	object
description	내부적으로 augmentObject() 호출 상속(inheritance)의 개념이 아닌 어떤( <i>r</i> ) 클래스의 prototype에 또다른 하나( <i>s</i> )의 prototype을 public method와 member를 추가합니다.
usage	YAHOO.lang.argumentProto(Foo, commonMethods, eventProvider);

# YAHOO.lang.argumentProto()

- augmentProto: `function(r, s) {`
- `if (!s||!r) {`
- `throw new Error("Augment failed, verify dependencies.");`
- `}`
- `//var a=[].concat(arguments);`
- `var a=[r.prototype,s.prototype];`
- `for (var i=2;i<arguments.length;i=i+1) {`
- `a.push(arguments[i]);`
- `}`
- `YAHOO.lang.augmentObject.apply(this, a);`
- `}`

# Combining Simple Data Sets

- `YAHOO.lang.merge()`는 data object를 단일화 시키는 역할을 합니다.
- `Prototype.js` 의 `Hash` 의 `merge()` 메서드와 동일한 기능을 합니다.

# YAHOO.lang.merge()

method	Static Method YAHOO.lang.argumentProto();
arguments	argumentProto( <i>object</i> r, <i>object</i> s)
return	object
description	augmentObject(r, s, <b>true</b> ) ->
usage	<pre>var set1 = { group : "ALC" }; var set2 = { name: "rhio.kim", age : 19 }; var set3 = { name: "star", age : 20, real : "kang" };  var total = YAHOO.lang.merge(set1, set2, set3, { member: 5 } );</pre>

# YAHOO.lang.merge()

- merge: `function() {`
- `var o={}, a=arguments;`
- `for (var i=0, l=a.length; i<l; i=i+1) {`
- `YAHOO.lang.augmentObject(o, a[i], true);`
- `}`
- `return o;`
- `}`

# YAHOO.namespace()

method	Static Method YAHOO.namespace();
arguments	namespace()
return	a reference to the last namespace object created
description	Argument 를 오브젝트화 하여 계층 구조를 만들고 마지막 Argument의 오브젝트 reference 를 반환합니다. 이미 존재하거나 ECMAScript 에서 제공된 예약어는 지원되지 않습니다. (e.g namespace('really.long.names'); )
usage	<pre>YAHOO.namespace("property.package"); YAHOO.namespace("YAHOO.property.package");  YAHOO.namespace("ALC", "BLC", "CLC.database"); //-&gt; return CLC.database object reference</pre>

# YAHOO.namespace()

- YAHOO.namespace = **function**() {
- **var** a=arguments, o=null, i, j, d;
- **for** (i=0; i<a.length; i=i+1) {
- d=a[i].split(".");
- o=YAHOO;
- *//YAHOO is implied, so it is ignored if it is included*
- **for** (j=(d[o] == "YAHOO") ? 1 : 0; j<d.length; j=j+1) {
- o[d[j]]=o[d[j]] || {};
- o=o[d[j]];
- }
- }
- **return** o;
- };

# YAHOO.log()

method	Static Method YAHOO.log();
arguments	log( <i>string</i> msg, <i>string</i> cat, <i>string</i> src)
return	boolean
description	YAHOO.widget.Logger 을 이용하여 위젯 형식의 로그 메시지를 볼 수 있다
usage	none



# YAHOO.log()

- `YAHOO.log = function(msg, cat, src) {`
- `var l=YAHOO.widget.Logger;`
- `if(l && l.log) {`
- `return l.log(msg, cat, src);`
- `} else {`
- `return false;`
- `}`
- `}`

# YAHOO.register()

method	Static Method YAHOO.register();
arguments	register( <i>string</i> name, <i>function</i> mainClass, <i>object</i> data)
return	none
description	namespace() 유사한 기능으로 개발자에 의해 생성된 모듈을 YAHOO.env.modules 에 등록합니다.
usage	<pre>var YAHOO.example.alc = {}; YAHOO.register('blc', YAHOO.example.alc, { version : '0.1', build : 152 });</pre>

# YAHOO.register()

```
YAHOO.register = function(name, mainClass, data) {  
    var mods = YAHOO.env.modules;  
    if (!mods[name]) {  
        mods[name] = { versions:[], builds:[] };  
    }  
    var m=mods[name],v=data.version,b=data.build,ls=YAHOO.env.listeners;  
    m.name = name;  
    m.version = v;  
    m.build = b;  
    m.versions.push(v);  
    m.builds.push(b);  
    m.mainClass = mainClass;  
    // fire the module load listeners  
    for (var i=0;i<ls.length;i=i+1) {  
        ls[i](m);  
    }  
    // label the main class  
    if (mainClass) {  
        mainClass.VERSION = v;  
        mainClass.BUILD = b;  
    } else {  
        YAHOO.log("mainClass is undefined for module " + name, "warn");  
    }  
}
```