1. $A 클래스
   * 배열을 다양하게 조작할수 있는 메소드를 제공하는 wrapper class입니다.
   * properties.
   * methods
     + 생성자
       - 인수로 들어온 Array 객체를 가지고 새로운 $A객체를 만들어냅니다.
       - Array arrary
         1. 인자 array가 배열일 경우 해당 배열을 private 변수인 \_array에 저장합니다. 없을 시 새로운 배열을 만듭니다.
     + String toString()
       - 배열을 나타내는 문자열을 리턴합니다.
       - return String
         1. 배열을 나타내는 문자열을 리턴합니다.
     + Number length(Number nLen, Object oValue)

배열의 길이를 변경하고, 증가 시 원하는 값으로 초기화할 수 있습니다.

* + - * Number nLen
        1. 배열의 길이를 리턴하는 인자입니다. \_array 배열의 길이보다 클 경우, \_array 배열의 크기를 nLen만큼 증가시키고 oVlaue로 채웁니다.
      * Object oValue
        1. 배열의 길이를 늘리려고 할 경우, 늘어가는 배열에 들어가는 값입니다.
      * return Number
        1. 변경된 배열의 길이를 리턴합니다.
    - Boolean has(Object oValue)

특정값이 배열에 있는지 확인합니다..

* + - * Object oValue
        1. 배열에 있는 지 확인하고 싶은 변수입니다.
      * return Boolean
        1. 배열에 oValue가 있으면 true,없으면 false를 반환합니다.
    - Number indexOf(Object oValue)
      * 원하는 값이 있는 인덱스를 가져옵니다.
      * oValue
        1. 배열에 있는지 확인하고 싶은 변수입니다.
      * return Number
        1. 배열안에서 원하는 값이 있는 인덱스입니다.
    - Array value()
      * 자기 자신의 값을 알 수 있습니다.
      * return Array
        1. 프로퍼티 \_array를 리턴합니다..
    - Number push(Object oValue)
      * 특정 값을 배열의 맨 끝에 추가합니다..
      * Object oValue
        1. 배열에 넣고 싶은 값
      * return Number
        1. 바뀐 배열의 길이입니다.
    - Object pop()
      * 배열의 맨 끝 원소를 가져옵니다. 배열 안에서 해당원소는 삭제됩니다..
      * return Object
        1. 배열의 맨끝에 위치한 원소입니다.
    - Object shift()
      * 배열안의 원소를 앞으로 한칸씩 이동시킵니다. 맨 앞 원소는 삭제됩니다.
      * return Object
        1. 배열의 맨 앞에 있던 원소입니다.
    - Number unShift(/\* oValue1…\*/)
      * 배열의 맨앞에 원하는 값(들)을 삽입합니다.
      * Object oValue1 ..
        1. 배열에 추가하기 원하는 값입니다.
      * return Number
        1. 바뀐 배열의 길이입니다.
    - Obejct forEach(Function fCallback, Object oThis)
      * fCallback함수를 배열을 순환하며 각 원소에 대해 실행합니다.
      * Function fCallback
        1. 배열의 각 원소에 실행될 콜백함수입니다.
      * Object oThis
        1. 콜백함수의 this 객체입니다.
      * return this
        1. 자기자신을 반환합니다.
    - $A slice(Number nStart, Number nEnd)
      * 배열의 일부분을 반환합니다 원래의 배열은 변하지 않습니다..
      * Number nStart
        1. 잘라내기를 시작할 인덱스 값입니다.음수일시 뒤에서 부터의 위치를 나타냅니다.
      * Number nEnd
        1. 잘라내기를 마칠 인덱스 값입니다. 음수일시 뒤에서부터의 위치를 나타내고, 없을 시 배열의 끝까지 잘라냅니다.
      * return $A
        1. 원본 배열에서 가져온 일부분입니다.
    - $A splice(Number nStart, Number nHowMany, /\*oValue1…\*/)
      * 배열의 원소를 삭제하거나 추가합니다. 배열 원소의 삭제가 일어났을 시 삭제된 원소들의 배열을 반환합니다.
      * Number nStart
        1. 추가 또는 삭제할 배열 원소의 시작위치
      * Number nHowMany
        1. 배열에서 삭제하고 싶은 원소의 개수입니다. 지정되지 않을 경우 nStart부터 끝까지 모든 원소를 삭제합니다.
      * oValue1…
        1. 배열에 추가하고 싶은 값들의 배열입니다.
      * return $A
        1. 삭제가 일어났을 시, 배열에서 삭제되는 원소들의 배열입니다.
    - $A shuffle()
      * 배열의 원소를 랜덤하게 섞습니다.
      * return this
        1. 자기자신을 반환합니다.
    - $A reverse()
      * 배열 원소들의 순서를 뒤집습니다.
      * return this
        1. 자기 자신을 반환합니다.
    - $A empty()
      * 배열내의 모든 원소를 삭제하고 빈 배열로 만듭니다.
      * return $A
        1. 빈 배열(this)을 반환합니다.
    - void Break()
      * 해당 제어블록을 빠져나갑니다.
    - void Continue()
      * 제어블록 안에서 자신 밑의 명령행을 수행하지 않습니다.
    - $A map(Function fCallback, Object oThis)
      * 배열을 순환하며 각 원소에 대해 콜백함수를 실행합니다.
      * Function fCallback
        1. 배열의 각 원소에 실행될 콜백함수입니다.
      * Object oThis
        1. 콜백함수의 this 객체입니다.
      * return Obejct
        1. 자기자신을 반환합니다
    - $A filter(Function fCallback, oThis)
      * 배열을 순회하며 각각의 원소에 fCallback 함수를 적용시킵니다. 콜백함수를 만족시키는 원소는 새로운 배열에 담기게 됩니다.
      * Function fCallback
        1. 배열의 각 원소에 적용할 함수. Boolean 형태의 리턴값이 있어야한다.
      * oThis
        1. 콜백함수내에서의 this 객체입니다.
      * return $A
        1. 콜백함수를 만족하는 원소들로 이루어진 새로운 배열입니다.
    - Boolean every(Function fCallback, Object oThis)
      * 배열내의 모든 원소들이 fCallback 함수를 만족하는지 검사합니다.
      * Function fCallback
        1. 배열의 각 원소에 적용될 콜백함수입니다.
      * Object oThis
        1. 콜백함수의 this 객체입니다.
      * return Boolean
        1. 모든 원소가 만족하면 true를, 하나라도 만족하지 못하면 false를 반환합니다.
    - Boolean some(Function fCallback , Objct oThis)
      * 배열내의 원소중에 하나라도 fCallback 함수를 만족하는 원소가 있는지 검사합니다.
      * Function fCallback
        1. 배열의 각 원소에 적용될 콜백함수입니다.
      * Object oThis
        1. 콜백함수의 this객체입니다.
      * return Boolean
        1. 배열내에서 fCallback을 만족하는 원소가 하나라도 있으면 true, 하나도 없으면 false를 리턴합니다.
    - $A refuse(Object oValue1…)
      * 현재 배열에서 특정 값들을 제외한 배열을 가지고 옵니다
      * Obejcet oValue1 …
        1. 배열에서 제외하고 싶은 값들입니다.
      * return $A
        1. 현재 배열에서 인자로 들어온 값들을 제외한 새로운 배열입니다.
    - $A unique()
      * 배열에서 중복되는 값을 제거합니다.
      * return $A
        1. 중복되는 값을 제거한 배열 입니다.

1. $Element 클래스
   * 주어진 DOM엘리먼트를 조작할수있는 다양한 메서드를 제공하는 wrapper class입니다.
   * properties
     + Boolean IS\_IE
       - 웹브라우져가 Internet Explore인지 확인하는 변수입니다.
     + Boolean IS\_FF
       - 웹브라우져가 FireFox 인지 확인하는 변수입니다.
     + Boolean IS\_OP
       - 웹브라우져가 Opera인지 확인하는 변수입니다.
     + Boolean IS\_SF
       - 웹브라우져가 Safari인지 확인하는 변수입니다.
     + Boolean IS\_CH
       - 웹브라우져가 Chrome인지 확인하는 변수입니다.
   * methods
     + 생성자
     + HTTPElement $value()
       - 클래스의 값(HTTPElement)를 반환합니다.
     + $Element show()
       - 엘리먼트가 화면에 보이도록 display설정을 변경합니다. 미리 설정된 기본값이 없을 경우에는 inline으로 설정합니다. 에러가 발생시 block으로 설정됩니다.
       - return this
         1. 변경된 설정이 적용된 HTTPElement입니다.
     + $Element hide()
       - 엘리먼트가 화면에 보이지 않도록 display설정을 변경합니다.
       - return this
         1. 설정이 변경된 엘리먼트입니다.
     + Boolean visible(Boolean bVisible)
       - 전달된 인자가 true이면 화면에 나타나도록, false이면 화면에서 보이지 않도록 엘리먼트의 속성을 변경합니다.인자가 undefined시에는 해당 엘리먼트의 display 속성이 none인지 아닌지를 판단하여 반환합니다.
       - return Boolean
         1. 해당 엘리먼트의 display 속성이 none아 아닐경우 true를, 그 외에는 false를 리턴합니다.
     + $Element toggle()
       - 해당 엘리먼트를 화면에 나타낼 지 여부를 변경합니다.
     + Number opacity(value)
       - 엘리먼트의 투명도를 설정합니다. filter가 없다면 visible 상태여부를 판단하여 100 혹은 0의 값을 주게 됩니다.
       - Number value
         1. 투명도를 설정할 값입니다. 0~1사이를 초과할 시 0 또는 1로 설정됩니다.
       - return Number
         1. 설정한 투명도 값입니다.
     + null | Object \_getCss(HTTPElement e, String sName)
       - 원하는 css 값을 얻어옵니다
       - HTTPElement e
         1. css 값을 알고 싶은 엘리먼트입니다.
       - String sName
         1. 알고 싶은 css 값의 종류입니다.
     + Object css( String|Array sName, String sValue)
       - css 값을 얻어오거나 설정합니다.
       - String | Array sName
         1. 알고싶은 css값의 속성명입니다.
         2. 값을 설정할때에는 배열로 넘겨줄 수 있습니다.
       - return null | $Element
         1. 찾고자 하는 값이 없을 시에는 null을 반환합니다.
     + String | $Element attr(String |Object sName, String sValue)
       - 엘리먼트에 attribute를 가져오거나 설정할 수 있습니다.
       - String | Object sName
         1. 가져오거나 설정하고 싶은 엘리먼트의 속성 이름 입니다. 이 인자만 사용해서 호출 시 해당하는 값을 가져오게 되고, 두번째 인자까지 전달시 이 이름에 해당하는 속성값을 설정할 수 있습니다. 해당 속성값의 이름은 배열 혹은 $H 객체 일수 있습니다.(여러개 설정시)
       - String | Number sValue
         1. 설정한 속성 값입니다. null 일경우 해당 속성을 삭제합니다.
       - return String | $Element
         1. 가져온 속성 값입니다. 속성 값을 설정하였을 때에는 $Element를 리턴합니다.
     + Number | $Element width(width)
       - 엘리먼트의 width값을 가져오거나 설정합니다.
       - Number width
         1. 설정하고자 하는 값으로 단위는 px입니다.
       - return Number | $Element
         1. 엘리먼트의 width를 반환합니다. 설정시에는 this obj를 반환합니다.
     + Number | $Element height(height)
       - 엘리먼트의 높이를 가져오거나 설정합니다.
       - Number height
         1. Number type일 경우 높이를 설정합니다.height에 설정할 높이값입니다. 단위는 px입니다.
       - return Number | $Element
         1. 엘리먼트의 높이입니다. 설정시에는 this obj를 반환합니다.
     + String | $Element className(String sClass)
       - 엘리먼트의 클래스 이름을 가져오거나 설정합니다.
       - String sClass
         1. 엘리먼트에 설정할 클래스이름입니다. 이 인자 없이 호출 시 엘리먼트의 현재 클래스이름을 반환합니다.
       - return String | $Element
         1. 클래스 이름 설정시 엘리먼트를 반환하고 그 외에는 엘리먼트의 클래스 이름을 반환합니다.
     + Boolean hasClass (String sClass)
       - 엘리먼트가 특정 클래스 이름을 사용하고 있는 확인합니다..
       - String sClass
         1. 확인할 클래스 이름입니다.
       - return Boolean
         1. 클래스의 사용 여부를 반환합니다.
     + $Element addClass (String sClass)
       - 엘리먼트에 클래스 이름을 추가합니다.
       - String sClass
         1. 추가할 클래스 이름입니다.
       - return $Element
         1. this obj를 반환합니다.
     + $Element removeClass (String sClass)
       - 엘리먼트의 특정 클래스 이름을 제거합니다.
       - String sClass
         1. 제거할 클래스 이름입니다.
       - return $Element
         1. this obj를 반환합니다.
     + void toggleClass (String sClass1,String sClass2)
       - 클래스를 토글합니다. 첫번째 인수가 들어왔을 시에는 해당 클래스만 트글하고 두개를 전달하여 호출하면 설정된 것을 지우고 나머지를 추가합니다.
       - String sClass
         1. 토글할 클래스 이름입니다.사용여부를 확인하여 토글합니다.
       - String sClass2
         1. 위의 인자를 포함하여 두개중 사용중인 것을 지우고 새로운 클래스를 설정합니다.
     + String | $Element text(String sText)
       - 엘리먼트의 텍스트 값을 가져오거나 설정합니다.
       - String sText
         1. 설정할 인자값입니다. 생략시 해당 엘리먼트의 텍스트 노드 값들 반환하게 됩니다.
       - return String | $Element
         1. 설정 시 this obj를, 가져올시에는 가져온값을 반환합니다.
     + String html(sHTML)
       - 엘리먼트가 innerHTML 값을 가져오거나 설정합니다.
       - String sHTML
         1. 설정할 HTML문자열입니다. 생략시 해당 엘리먼트의 innerHTML 값을 가져옵니다.
       - return String
         1. 해당 엘리먼트의 innerHTML값입니다.
     + String outerHtml()
       - 엘리먼트가 outerHTML 값을 가져옵니다. 최상위 노드의 경우 innerHTML을 반환합니다.
       - return String
         1. 해당 엘리먼트의 outerHTML값입니다.
     + String toString()
       - 해당 엘리먼트를 나타내는 문자열을 반환합니다. outerHTML과 그 반환값이 동일합니다.
       - return String
         1. 해당 엘리먼트의 outerHTML값입니다.
     + $Element appear(Number duration, Function callback)
       - 엘리먼트에 fade in효과를 줍니다.
       - Number duration
         1. 엘리먼트의 투명도가 현재부터 1이될때까지 걸리는 시간입니다.단위는 sec입니다.
       - Function callback
         1. 투명도가 1이 된후에 실행할 함수입니다.
       - return $Element
         1. this obj 를 반환합니다.
     + $Element evalScripts(String sHTML)
       - <script>태그 안에 있는 문자열을 파싱해서 eval로 실행합니다.
       - String sHTML
         1. <script>가 들어있는 문자열입니다.
       - return $Element
         1. this obj를 반환합니다.
     + $Element append(Obect oElement)
       - 엘리먼트에 마지막 자식 노드를 추가합니다.
       - Object oElement
         1. 엘리먼트에 추가할 자식 엘리먼트입니다.
       - return String
         1. 해당 엘리먼트의 outerHTML값입니다.
     + Element prepend(Obect oElement)
       - 엘리먼트에 맨 앞에 자식 노드를 추가합니다.
       - Object oElement
         1. 엘리먼트에 추가할 자식 엘리먼트, 혹은 id 값입니다.
       - return String
         1. 해당 엘리먼트의 outerHTML값입니다.
     + $Element replace(Obect oElement)
       - 엘리먼트에 노드를 인수로 지정한 엘리먼트로 대체합니다..
       - Object oElement
         1. 대체할 엘리먼트입니다. 엘리먼트 id가 올수도 있습니다.
       - return $Element
         1. 엘리먼트가 대체된 this obj를 리턴합니다.
     + $Element appendTo(Obect oElement)
       - 엘리먼트에 마지막 자식 노드로 인수로 들어온 엘리먼트를 추가합니다.
       - Object oElement
         1. 추가할 엘리먼트입니다. 엘리먼트 id가 올수도 있습니다.
       - return $Element
         1. 엘리먼트가 추가된 this obj를 리턴합니다.
     + $Element prependTo(Obect oElement)
       - 엘리먼트에 맨앞 자식 노드로 인수로 들어온 엘리먼트를 추가합니다.
       - Object oElement
         1. 추가할 엘리먼트입니다. 엘리먼트 id가 올수도 있습니다.
       - return $Element
         1. 엘리먼트가 추가된 this obj를 리턴합니다.
     + HTTPElement before(HTTPElement oElement)
       - 엘리먼트의 바로 앞에 인수로 들어온 엘리먼트를 삽입합니다.
       - Object oElement
         1. 삽입할 엘리먼트입니다. 엘리먼트 id가 올수도 있습니다.
       - return HTTPElement
         1. 삽입한 엘리먼트를 반환합니다.
     + $Element | Array parent(Function pFunc, Number limit)
       - 엘리먼트에 부모 노드를 검색합니다.
       - Function pFunc
         1. 상위 노드의 검색 조건을 지정해 놓은 콜백함수입니다. 다수의 상위 노드를 조건 없이 검색시에는 null로 지정하고 limit에 검색할 상위노드의 갯수를 설정합니다.
       - Number limit
         1. 탐색할 상위노드의 갯수입니다.
       - return $Element | Array
         1. 하나의 상위노드만 찾을 시에는 $Element를 반환하고, 여러개의 상위노드를 검색시에는 배열로 반환합니다.
     + $Element | Array child(Function pFunc, Number limit)
       - 엘리먼트에 자식 노드를 검색합니다.
       - Function pFunc
         1. 자식 노드의 검색 조건을 지정해 놓은 콜백함수입니다. 다수의 자식 노드를 조건 없이 검색시에는 null로 지정하고 limit에 검색할 자식노드의 갯수를 설정합니다.
       - Number limit
         1. 탐색할 자식노드의 갯수입니다.
       - return $Element | Array
         1. 하나의 자식노드만 찾을 시에는 $Element를 반환하고, 여러개의 자식노드를 검색시에는 배열로 반환합니다.
     + $Element | Array prev(Function pFunc)
       - 특정노드 이전에 나오는 형제 노드를 검색합니다.
       - Function pFunc
         1. 형재 노드의 검색 조건을 지정해 놓은 콜백함수입니다. 생략시 바로전의 형제노드를 반환하게 됩니다.
       - return $Element | Array
         1. 하나의 형제노드만 찾을 시에는 $Element를 반환하고, 여러개의 형제노드를 검색시에는 배열로 반환합니다.
     + $Element | Array next(Function pFunc)
       - 특정노드 다음에 나오는 형제 노드를 검색합니다.
       - Function pFunc
         1. 형재 노드의 검색 조건을 지정해 놓은 콜백함수입니다. 생략시 바로전의 형제노드를 반환하게 됩니다.
       - return $Element | Array
         1. 하나의 형제노드만 찾을 시에는 $Element를 반환하고, 여러개의 형제노드를 검색시에는 배열로 반환합니다.
     + $Element first()
       - 엘리먼트의 처음 자식 노드를 가져옵니다.
       - return $Element
         1. 처음 자식 노드를 반환합니다.
     + $Element last()
       - 엘리먼트의 마지막 자식 노드를 가져옵니다.
       - return $Element
         1. 마지막 자식 노드를 반환합니다.
     + Boolean isChildOf(Object element)
       - 엘리먼트가 인수로 들어온 엘리먼트의 자식노드인지 확인합니다.
       - Object element
         1. 부모인지 확인하고 싶은 엘리먼트 혹인 엘리먼트의 id값입니다.
       - return Boolean
         1. 자식 여부를 반환합니다.
     + Boolean isParentOf(Object element)
       - 엘리먼트가 인수로 들어온 엘리먼트의 부모노드인지 확인합니다.
       - Object element
         1. 자식인지 확인하고 싶은 엘리먼트 혹인 엘리먼트의 id값입니다.
       - return Boolean
         1. 부모 여부를 반환합니다.
     + Boolean isEqual(Object element)
       - 엘리먼트가 인수로 들어온 엘리먼트와 같은 엘리먼트인지 확인합니다.
       - Object element
         1. 비교할 엘리먼트 입니다.
       - return Boolean
         1. 일치 여부를 반환합니다.
     + $Element fireEvent(String sEvent)
       - 엘리먼트에 이벤트를 발생시킵니다.
       - String sEvent
         1. 발생시킬 이벤트의 이름입니다. on을 생략하고 사용합니다.
       - return $Element
         1. this obj를 반환합니다.
     + $Element empty()
       - 엘리먼트에 모든 자식노드들을 제거합니다.
       - return $Element
         1. this obj를 반환합니다.
     + $Element remove(Object oChild)
       - 엘리먼트에 특정 자식노드들을 제거합니다.
       - Object oChild
         1. 제거할 자식 엘리먼트입니다.
       - return $Element
         1. this obj를 반환합니다.
     + $Element leave()
       - 엘리먼트를 그 자신의 부모로 부터 제거합니다.
       - return $Element
         1. 제거한 $Element 객체
     + $Element wrap(String|HTTPElement wrapper)
       - 엘리먼트를 인수로 들어온 엘리먼트로 감쌉니다.
       - String | HTTPElement
         1. 감쌀 엘리먼트 혹은 감쌀 엘리먼트의 id값입니다.
       - return $Element
         1. 새로운 노드로 감싸진 $Element 객체
     + Number indexOf($Element element)
       - 엘리먼트의 자식 노드들 중에 인수로 들어온 엘리먼트가 몇번째 인지 알아옵니다. 없을 시 에는 -1을 반환합니다.
       - $Element element
         1. 몇번째 자식인지 알고싶은 엘리먼트
       - return Number
         1. 0번부터 시작한 인수로 들어온 엘리먼트가 현재 엘리먼트의 자식 노드들중에서 몇번인지를 나타내는 값입니다. 없을시 -1을 반환합니다.
2. $Fn
   * 함수 객체를 조작하고 생성하는 클래스입니다.
   * properties
   * methods
     + 생성자
     + 함수,혹은 함수를 나타내는 문자열을 받아 $Fn객체를 생성합니다.
       - Function | String func
         1. $Fn클래스로 감쌀 함수객체입니다. 혹은 함수의 인자를 나타내는 문자열일수도 있습니다.
       - Object | String thisObject
         1. 함수객체가 다른 객체의 메소드일때 해당 객체도 같이 전달합니다. 혹은 함수 몸체를 나타내는 문자열일수도 있습니다.
     + Function value()
       - 원래의 Function 객체를 반환합니다.
       - return Function
         1. \_func에 설정되어 있는 Function 객체를 반환합니다.
     + Function bind()
       - 객체와 메소드를 하나로 묶어줍니다.
       - return Function
         1. 메소드로 묶인 Function 객체입니다.
     + Function bindForEvent()
       - 객체와 메소드를 하나로 묶어 하나의 Event Handler Function으로 만들어줍니다.
       - return Function
         1. 메소드로 묶인하나의 Event Handler Function입니다.
     + $Fn attach(HTTPElement oElement, String sEvent)
       - 함수를 특정 엘리먼트의 이벤트 핸들러로 등록합니다.
       - HTTPElement oElement
         1. 이벤트를 추가할 엘리먼트입니다.
       - String sEvent
         1. 추가할 이벤트의 종류를 나타내는 문자열입니다.
       - return $Fn
         1. 추가한 $Fn객체를 리턴합니다.
     + $Fn attach(HTTPElement oElement, String sEvent)
       - 함수를 특정 엘리먼트의 이벤트 핸들러에서 제거합니다.
       - HTTPElement oElement
         1. 핸들러를 제거할 엘리먼트입니다.
       - String sEvent
         1. 제거할 이벤트의 종류를 나타내는 문자열입니다.
       - return $Fn
         1. 제거한 $Fn객체를 리턴합니다.
     + $Fn delay(Number nSec, Array args)
     + 지정한 시간이후에 지정한 인수로 함수를 호출합니다.= setTimeout
       - Number nSec
         1. 대기할 시간입니다.단위는 Sec입니다.
       - Array args
         1. 함수로 전달할 인자들의 배열입니다.
       - return $Fn
         1. this obj를 리턴합니다.
     + Number setInterval(Number nSec,Array args)
       - 기존 setInterval 함수와 동일하게 작동합니다
       - Number nSec
         1. 실행주기를 설정하는 값입니다. 단위는 Sec입니다.
       - Array args
         1. 함수에 전달할 인자들의 배열입니다.
       - return Number
         1. clearTimeout을 하기위한 timeOutId값 입니다.
     + void free(HTTPElement oElement)
       - 지정된 엘리먼트에 등록된 모든 이벤트 핸들러를 제거합니다.
       - HTTPElement oElement
         1. 이벤트를 제거할 엘리먼트입니다.
     + void gc()
       - 내부에 있는 pool 객체에 등록되어 있는 이벤트 핸들러들을 제거합니다.window가 unload될때에도 이벤트핸들러들을 제거합니다.