

## window.document.TextRange Objekt des Internet Explorer

**window.document.TextRange Objekt des Internet Explorer.....1**  
**window.document.TextRange.TextRectangle Collection des Internet Explorer ..... 3**  
**window.document.TextRange.TextRectangle Objekt des Internet Explorer..... 4**

Objekt des gesamten Plaintextes eines Objektes (Textbereich, Textrange)

basiert auf dem Objekt TextNode

Elternobjekte mit Textrange sind alle Objekte, die eine der nachfolgend beschriebenen Methoden besitzen

- z.B. body Objekt
  - button Objekt
  - textarea Objekt
  - input text Objekt
  - selection Objekt (nur wenn ein Text selektiert wurde (mit oder ohne HTML))
- können weitere HTML-Elemente enthalten, die ebenfalls Textbereiche besitzen können

Textbereich kann in Textteile zerlegt sein: Jeder Teil ist ein Element mit Plaintext.  
hat seinen Bereichrahmen (siehe Objekt TextRange.TextRectangle und Collection TextRange.TextRectangle)

**Erzeugung:**

**in HTML** innerhalb der Tag-Begrenzer muss Text kodiert sein  
z.B. leerer DIV hat keinen Textbereich  
kann nachträglich einen Textbereich per Script erhalten

```
Beispiel: <HTML>
<BODY>
<H1>Hallo</H1>
<CENTER><H2> und willkommen ! </H2></CENTER>
</BODY>
</HTML>
```

Textrange ist erzeugt worden im BODY-Teil  
und enthält nur Plain-Text also  
"Hallo und willkommen !"  
hat genau einen Zeiger auf die Textrange-Element

Textrange-Element ist z.B. das Wort "Hallo" des H1-Elementes im Container BODY

**per Script:**

.createTextRange() Textbereich erzeugen  
Syntax:  
[ var Zeiger = ] object.createTextRange()

Zeiger auf Range  
null wenn TextRange nicht erzeugbar

Beispiel 1:

```
<SCRIPT LANGUAGE="JScript">
var FeldAllerButtonElemente coll = document.all.tags("BUTTON");

if ( (FeldAllerButtonElemente !=null )
&& (FeldAllerButtonElemente.length>0)
)
{
var ZeigerAufRange = FeldAllerButtonElemente[0].createTextRange();
ZeigerAufRange.text = "Clicked";
}
</SCRIPT>
```

Beispiel 2:

```
function ErzeugeUndSelektiereTextRange()
{
var TextBereich = document.body.createTextRange();
TextBereich.findText("Testtext");
TextBereich.select();
}
```

.createRange() Zeiger auf einen Universal-Bereich erzeugen per document.selection Objekt des Dokumentes  
Universal-Bereich kann enthalten:  
Text (document.selection.textrange Collection  
textrange Objekt)  
oder Control-Element(e) (document.selection.controlRange Collection)  
siehe auch Methoden .createControlRange() .createTextRange() und .createRangeCollection()



Eigenschaft .type

Syntax:

[ var Zeiger = ] document.selection.createRange()

.createRangeCollection()

document.selection.textrange Collection erzeugen per document.selection Objekt des Dokumentes nur wenn der Browser multiple Selektion unterstützt, so mehr als 1 Feld-Element textrange Objekt vorhanden bei Mehrfach-Selektion durch User

siehe auch textrange Objekt

Methoden .createRange() .createControlRange() und .createTextRange()

Syntax:

[ var Zeiger = ] = document.selection.createRangeCollection()

**Eigenschaften:**

.boundingHeight

Höhe des Rechteckes in Pixel um den Textbereich des TextRange Objektes per textrange Objekt

.boundingLeft

Abstand linker Rand des Rechteckes in Pixel um den Textbereich zur linkem Rand des Container-Objektes, in dem der Textbereich liegt

per textrange Objekt

.boundingTop

Abstand oberer Rand des Rechteckes in Pixel um den Textbereich zum oberen Rand des Container-Objektes, in dem der Textbereich liegt

per textrange Objekt

.boundingWidth

Breite des Rechteckes in Pixel um den Textbereich des TextRange Objektes

per textrange Objekt

.htmlText

HMTL-Text im Textbereich nicht den Plaintext-Anteil liefern

per textrange Objekt

nur unter Windows 32-Bit

.offsetLeft

X-Koordinate der linken oberen Ecke Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)

.offsetTop

Y-Koordinate der linken oberen Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)

.text

Plain-Text im Textbereich per textrange Objekt Dokument muss komplett geladen sein nur unter Windows 32-Bit

**Methoden:**

.collapse()

Textbereich-Zeichen-Zeiger auf Anfang oder Ende des Textbereiches setzen

Hinweis: Zeiger nur auf Plain-Text-Zeichen

Bsp.: <BODY><P><B>abc

Zeiger kann wandern von VOR a bis HINTER c

VOR a entspricht Start des Bereiches mit Zeigerwert 0

HINTER c entspricht Ende des Bereiches mit Zeigerwert 3

per textrange Objekt

nur unter Windows 32-Bit

.compareEndpoints()

Vergleich der Textbereich-Zeichen-Zeiger von 2 Textbereichen

Hinweis: Zeiger nur auf Plain-Text-Zeichen

Bsp.: <BODY><P><B>abc

Zeiger kann wandern von VOR a bis HINTER c

VOR a entspricht Start des Bereiches mit Zeigerwert 0

HINTER c entspricht Ende des Bereiches mit Zeigerwert 3

per textrange Objekt

nur unter Windows 32-Bit

.duplicate()

Zeiger auf eine Duplikat-Instanz eines Textbereiches liefern

per textrange Objekt

nur unter Windows 32-Bit

Prüfung eines Textbereichen auf Kopie eines anderen Textbereiches per Methode .inRange() bzw. .isEqual()

.execCommand()

Kommando ausführen z.B. im aktuellen Dokument in aktueller Selektion im aktuellen Bereich

erst nach dem kompletten Laden des Dokumentes zulässig

Hinweis: Selektion = Markierung z.B. von Textbereich (Block)

Control = Element zur Steuerung analog zum HTML-Element (Tag)

Input-Control = Element mit Eingabeeigenschaft

.expand()

Plain-Text als Teil eines Textbereiches derart ausdehnen, dass er komplett die Dimension des Elternobjektes (Container-Objektes) einnimmt

per textrange Objekt

nur unter Windows 32-Bit

.findText()

Text im gesamten Textbereich suchen

per textrange Objekt

nur unter Windows 32-Bit



.getBookmark()	für Nutzung einer Textmarke siehe Methoden .getBookmark() und .moveToBookmark() Textmarke (Bookmark) im Textbereich setzen Textmarke kann mit Methode .moveToBookmark() anpositioniert werden per textrange Objekt nur unter Windows 32-Bit
.getBoundingClientRect()	Referenz auf TextRectangle-Objekt im Element holen
.getClientRects()	Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster Feld mit Index als Integer ab 0 pro Eintrag ein Rectangle prüfen ob 2 Textbereiche sich einschliessen z.B. bei verschachtelten DIV's
.inRange()	prüfen ob 2 Textbereiche sich einschliessen z.B. bei verschachtelten DIV's per textrange Objekt nur unter Windows 32-Bit
.isEqual()	Auf Identität zweier Textbereiche prüfen per textrange Objekt
.move()	Textbereich-Zeichen-Zeiger bewegen bezüglich aktueller Position im Textbereich per textrange Objekt nur unter Windows 32-Bit
.moveEnd()	Textbereich-Ende neu setzen (für den Textbereich-Zeiger) bezüglich aktueller Position im Textbereich per textrange Objekt nur unter Windows 32-Bit
.moveStart()	Textbereich-Anfang neu setzen (für den Textbereich-Zeiger) bezüglich aktueller Position im Textbereich per textrange Objekt nur unter Windows 32-Bit
.moveToBookmark()	Textbereich-Zeichen-Zeiger auf eine Textmarke (Bookmark) im Textbereich positionieren Textmarke wurde gesetzt per Methode .getBookmark() per textrange Objekt nur unter Windows 32-Bit
.moveToElementText()	Textbereich in ein Element/Objekt bewegen, das Text enthalten darf per textrange Objekt nur unter Windows 32-Bit
.moveToPoint()	Inhalt des Textbereiches um eine Pixelspanne verschieben (Offset) relativ zur linken oberen Fensterecke nach Verschiebung kann Textbereich leer sein per textrange Objekt nur unter Windows 32-Bit
.parentElement()	Zeiger auf das Elternelement (Container) des Textbereiches liefern Hinweis bei Elementverschachtelung: es wird das direkt um den Textbereich liegende Element referenziert per textrange Objekt nur unter Windows 32-Bit
.pasteHTML()	Textbereich-Inhalt durch Text ersetzen oder leeren Textbereich füllen Text kann auch HTML enthalten Tabelle nur mit kompletten HTML-Code einfügbar, also z.B. keine einzelne Zelle Achtung: Wenn HTML-Code im Text enthalten ist, muss das Elternobjekt auch diesen ansich unterstützen Bsp.: textArea erlaubt kein HTML-Code HTML-Code wird geparkt per textrange Objekt nur unter Windows 32-Bit
.queryCommandEnabled()	prüfen ob Kommando ausführbar ist
.queryCommandIndeterm()	prüfen ob Kommando-Status bestimmbar ist oder nicht
.queryCommandState()	Status des aktuellen Kommando ermitteln: ob ausgeführt wurde oder nicht
.queryCommandSupported()	prüfen ob Kommando im aktuellen Bereich unterstützt wird
.queryCommandValue()	Wert eines Kommandos liefern
.scrollIntoView()	Objekt derart scrollen, dass es im Fenster für User sichtbar wird Objekt muss an sich schon renderbar sein
.select()	Objekt selektieren z.B. Textbereich: wird hervorgehoben ControlRange: es wird Rechteck-Rahmen erzeugt per textrange Objekt nur unter Windows 32-Bit
.setEndPoint()	Textbereichanfang bzw. -ende von 2 Textbereichen synchronisieren per textrange Objekt nur unter Windows 32-Bit

## ***window.document.TextRange.TextRectangle Collection des Internet Explorer***

ab IE 5.x

Feld der TextRectangle-Objekte im HTML-Element (Objekt) mit Plain-Text

TextRectangle-Objekt: Rahmen der Positionierung einer einzelnen Textzeile im Plain-Text

Positionierung bezüglich Koordinatensystem des HTML-Dokumentes

Beispiel: Ein DIV enthält Plain-Text:

Jede im Browser dargestellte Zeile (auch jedes einzelne <BR>) besitzt einen eigenen Rechteck als



Rahmen der Positionierung innerhalb des DIV.  
Damit lassen sich Textelemente per Rahmen innerhalb des Dokumentes positionieren.

Achtung: Nach einer Änderung der Fensterdimension (resize) muss die Collection neu erzeugt werden (ist zu programmieren !).

**Syntax:**

```
[ var FeldZeiger = ] zeiger_auf_textrange.getClientRects();  
  
[ var FeldElementZeiger = ] FeldZeiger[Index];
```

Index: Integer und ab 0  
muss in [ ] kodiert sein

**Zugriff:**

```
FeldZeiger.eigenschaft  
FeldZeiger.methode()
```

**Eigenschaften:**

.length Anzahl der Feldelemente also Feldlänge z.B. bei Collection

**Methoden:**

```
.item() Referenz auf Feldelement anhand des Integer-Indexes oder des  
Attributnamen (analog zu ID oder NAME-Attribut) liefern  
außer bei Formular mit <INPUT TYPE=image ...>  
da dafür die children-Collection verwendet werden muss !!!  
.namedItem() Referenz auf Eintrag (FeldElement) anhand des Namen  
(analog zu ID oder NAME-Attribut) liefern
```

**window.document.TextRange.TextRectangle Objekt des Internet Explorer**

ab IE 5.x

TextRectangle-Objekt: Rahmen der Positionierung einer einzelnen Textzeile im Plain-Text (Positionierung bezüglich Koordinatensystem des HTML-Dokumentes)  
instanziert als Element der Collection TextRange.TextRectangle

Beispiel: Ein DIV enthält Plain-Text:

Jede im Browser dargestellte Zeile (auch jedes einzelne <BR>) besitzt einen eigenes Rechteck als Rahmen der Positionierung innerhalb des DIV.  
Damit lassen sich Textelemente per Rahmen innerhalb des Dokumentes positionieren.

Achtung: Nach einer Änderung der Fensterdimension (resize) muss die Collection neu erzeugt werden (ist zu programmieren !).

**Syntax:**

```
[ var Zeiger = ] zeiger_auf_textrectangle.getBoundingClientRect();  
  
zeiger_auf_textrectangle ist Element der Collection TextRange.TextRectangle  
  
[ var Zeiger = ] zeiger_auf_textrectangle_collection[Index].getBoundingClientRect();
```

Index Integer, ab 0  
muss in [ ] kodiert

Beispiel:

```
<HEAD>  
<SCRIPT>  
var ZeigerAufTextRectangleCollection;  
var Index =0;  
  
function Anzeigen(Zeiger) // Zeiger auf den einzigsten DIV mit Text also mit Textbereich  
{  
// aktuelle Collection der Rectangle von Div0 referenzieren:  
// Das ist nötig nach resize des Fensters, da resize leider nicht automatisch erkannt wird !  
ZeigerAufTextRectangleCollection = Zeiger.getClientRects();  
  
// Anzahl der Elemente ermitteln, also Anzahl der im Browser dargestellte Zeilen,  
// wobei es egal ist, ob diese per <BR> erzeugt wurden oder nicht  
// Hinweis: Zeilenlänge hängt auch von der Fensterbreite ab  
// (automatischer Umbruch)  
// Jede Zeile besitzt ihr eigenes Rectangle  
AnzahlTextRectangle = ZeigerAufTextRectangleCollection.length;  
  
// aktuellen Index prüfen ob letzte Zeile bereits erreicht wurde  
if (Index > AnzahlTextRectangle -1) // Index ab 1, Anzahl ab 1  
{  
// es wurde die letzte Zeile erreicht  
  
// Div2 unsichtbar machen also entfärben  
// Hinweis: Div2 liegt auf dem Rectangle von Div0  
ID_Div2.style.display="none";
```



```

// rücksetzen des Index, also mit erster Zeile weitermachen
Index = 0;
}

// Rechteck der aktuellen Zeile ermitteln unter Beachtung eines eventuellen Scrollens
var PosRechts = ZeigerAufTextRectangleCollection [Index].right + ID_Body.scrollLeft;
var PosLinks = ZeigerAufTextRectangleCollection [Index].left + ID_Body.scrollLeft;
var PosOben1 = ZeigerAufTextRectangleCollection [Index].top + ID_Body.scrollTop;

// und Div1 auf die Zeile positionieren, also Zeile einfärben
ID_Div1.style.top = PosOben1;
ID_Div1.style.width = (PosRechts - PosLinks) - 5;
ID_Div1.style.display = 'inline';

// aktuelle Position des Rechteckes von DIV0 ermitteln unter Beachtung eines eventuellen
// Scrollens
PosRechts = Zeiger.getBoundingClientRect().right + ID_Body.scrollLeft;
PosLinks = Zeiger.getBoundingClientRect().left + ID_Body.scrollLeft;
var PosOben2 = Zeiger.getBoundingClientRect().top + ID_Body.scrollTop;

// und Div2 überlagern positionieren
ID_Div2.style.top = PosOben2;
ID_Div2.style.width = (PosRechts - PosLinks) - 5;
ID_Div2.style.height = PosOben1 - PosOben2;

// aber nur einfärben, wenn mindestens 1 Zeile bereits eingefärbt wurde
if (Index > 0){ ID_Div2.style.display = 'inline';}

// Rectangle der nächsten Zeile einstellen
Index++;
}
</SCRIPT>
</HEAD>
<BODY ID="ID_Body">
  <DIV ID="ID_Div0"
  onclick=" Anzeigen(this)"
  >
    klicke
    <BR>
    ABCDEFGHIJKLMNOPQRSTUVWXYZ
    1234567890123456789012345678901234567890
    ABCDEFGHIJKLMNOPQRSTUVWXYZ
    1234567890123456789012345678901234567890
    ABCDEFGHIJKLMNOPQRSTUVWXYZ
    1234567890123456789012345678901234567890
    ABCDEFGHIJKLMNOPQRSTUVWXYZ
    123456789012345678901234567890
  </DIV>
  <DIV ID="ID_Div1"
  STYLE="position:absolute; left:5; top:400; z-index:-1; background-color:yellow; display:none"
  >
  </DIV>
  <DIV ID="ID_Div2"
  STYLE="position:absolute; left:5; top:400; z-index:-1; background-color:beige; display:none"
  >
  </DIV>
</BODY>

```

**Zugriff:**

Zeiger.eigenschaft

**Eigenschaften:**

sind les- und schreibbar

- .bottom
- .left
- .right
- .top

untere Pixelposition des Rechteckes um ein Objekt  
linke Pixelposition des Rechteckes um ein Objekt  
rechte Pixelposition des Rechteckes um ein Objekt  
obere Pixelposition des Rechteckes um ein Objekt

**Methoden:**

keine



- .bottom ..... 5
- .boundingHeight ..... 2
- .boundingLeft ..... 2
- .boundingTop ..... 2
- .boundingWidth ..... 2
- .collapse() ..... 2
- .compareEndpoints() ..... 2
- .createControlRange() ..... 1, 2
- .createRange() ..... 1, 2
- .createRangeCollection() ..... 1, 2
- .createTextRange() ..... 1, 2
- .duplicate() ..... 2
- .execCommand() ..... 2
- .expand() ..... 2
- .findText() ..... 2
- .getBookmark() ..... 3
- .getBoundingClientRect() ..... 3
- .getClientRects() ..... 3
- .htmlText ..... 2
- .inRange() ..... 3
- .isEqual() ..... 3
- .item() ..... 4
- .left ..... 5
- .length ..... 4
- .move() ..... 3
- .moveEnd() ..... 3
- .moveStart() ..... 3
- .moveToBookmark() ..... 3
- .moveToElementText() ..... 3
- .moveToPoint() ..... 3
- .namedItem() ..... 4
- .offsetParent ..... 2
- .offsetLeft ..... 2
- .offsetParent ..... 2
- .offsetTop ..... 2
- .parentElement() ..... 3
- .pasteHTML() ..... 3
- .queryCommandEnabled() ..... 3
- .queryCommandIndeterm() ..... 3
- .queryCommandState() ..... 3
- .queryCommandSupported() ..... 3
- .queryCommandValue() ..... 3
- .right ..... 5
- .scrollIntoView() ..... 3
- .select() ..... 3
- .setEndPoint() ..... 3
- .tags() ..... 4
- .text ..... 2
- .top ..... 5
- .type ..... 2
- ändern Textbereich-Zeichen-Zeiger ..... 3
- Anzahl der Feldelemente also Feldlänge ..... 4
- ausführen Kommando ..... 2
- bewegen Textbereich ..... 3
- body Objekt ..... 1
- Bookmark im Textbereich ..... 3
- Bookmark Textbereich ..... 3
- button Objekt ..... 1
- Collection document.selection.controlRange ..... 1
- Collection document.selection.textrange ..... 1, 2
- Collection TextRange.TextRectangle ..... 1, 3
- Control ..... 2
- createTextRange() ..... 1
- document.selection Objekt ..... 1, 2
- document.selection Objekt Selektion Universal-Bereich erzeugen ..... 1
- document.selection Objekt Universal-Bereich erzeugen Selektion ..... 1
- document.selection.controlRange Collection ..... 1
- document.selection.textrange Collection ..... 1, 2
- Dokument und Selektion ..... 1, 2
- duplizieren Textbereich ..... 2
- Elternelement des Textbereiches ..... 3
- Feld der Zeiger auf TextRectangle-Objekte ..... 3

- Feldelement Zeiger ..... 4
- Feldelemente Anzahl ..... 4
- Feldlänge ..... 4
- HTML-Text im Textbereich ..... 2
- HTML-Code im Textbereich ..... 3
- Inhalt Textbereich ersetzen ..... 3
- Inhalt Textbereich füllen ..... 3
- Inhalt Textbereich leeren ..... 3
- input text Objekt ..... 1
- Input-Control ..... 2
- Kommando Ausführbarkeit ..... 3
- Kommando ausführen ..... 2
- Kommando Ausführungserfolg ..... 3
- Kommando Unterstützung ..... 3
- Kommandos Wert ..... 3
- Kommando-Status ..... 3
- Koordinate der linken oberen Ecke Objektes ..... 2
- linke obere Ecke des Objektes ..... 2
- linke obere Ecke Objektes ..... 2
- Objekt body ..... 1
- Objekt button ..... 1, 2
- Objekt document.selection ..... 1, 2
- Objekt document.selection Selektion Universal-Bereich erzeugen ..... 1
- Objekt document.selection Universal-Bereich erzeugen Selektion ..... 1
- Objekt input text ..... 1
- Objekt linke obere Ecke ..... 2
- Objekt linke obere Ecke ..... 2
- Objekt Pixelposition des Rechteckes um Objekt ..... 5
- Objekt Referenz auf TextRectangle ..... 3
- Objekt scrollen ..... 3
- Objekt selection ..... 1
- Objekt selektieren ..... 3
- Objekt sichtbar ..... 3
- Objekt textarea ..... 1, 3
- Objekt TextNode ..... 1
- Objekt textrange ..... 1, 2, 3
- Objekt TextRange ..... 1
- Objekt TextRange.TextRectangle ..... 1, 4
- Objekt und sein Plaintext ..... 1
- Objekt Viewbereich ..... 3
- Objekt Zeiger auf TextRectangle ..... 3
- Pixelposition des Rechteckes um Objekt ..... 5
- Plaintext eines Objektes ..... 1
- Plain-Text im Textbereich ..... 2
- Plain-Text im Textbereiches dehnen ..... 2
- prüfen Textbereiche ..... 3
- Rechteck um Objekt Pixelposition ..... 5
- Rechteck um Textbereich ..... 2
- Rectangle ..... 3
- Referenz auf Feld der Zeiger auf TextRectangle-Objekte ..... 3
- Referenz auf Feldelement ..... 4
- Referenz auf TextRectangle-Objekt ..... 3
- selection Objekt ..... 1
- selektieren Objekt ..... 3
- Selektion ..... 2
- Selektion und Dokument ..... 1, 2
- Selektion Universal-Bereich erzeugen ..... 1
- suchen Text im Textbereich ..... 2
- Text im Textbereich suchen ..... 2
- Text positionieren ..... 3, 4
- textarea Objekt ..... 1, 3
- Textbereich ..... 1
- Textbereich bewegen ..... 3
- Textbereich Bookmark ..... 3
- Textbereich duplizieren ..... 2
- Textbereich Elternelement ..... 3
- Textbereich HTML-Text im Bereich ..... 2
- Textbereich Inhalt ersetzen ..... 3
- Textbereich Inhalt füllen ..... 3
- Textbereich Inhalt leeren ..... 3
- Textbereich mit HTML-Code ..... 3
- Textbereich Plain-Text dehnen ..... 2



Textbereich Plain-Text im Bereich ..... 2  
Textbereich Rechteck um Textbereich..... 2  
Textbereich Rechteckes um Textbereich ..... 2  
Textbereich Text suchen..... 2  
Textbereich Textmarke..... 3  
Textbereich Zeiger auf Elternelement..... 3  
Textbereiche prüfen..... 3  
Textbereich-Zeichen-Zeiger ..... 2  
Textlbereich-Zeichen-Zeiger ändern ..... 3  
Textmarke im Textbereich ..... 3  
Textmarke Textbereich..... 3  
TextNode Objekt ..... 1  
Textrange ..... 1

textrange Objekt ..... 1, 2, 3  
TextRange Objekt ..... 1  
TextRange.TextRectangle Collection ..... 1, 3  
TextRange.TextRectangle Objekt..... 1, 4  
TextRectangle..... 3, 4  
TextRectangle-Objekt ..... 3  
Textzeile positionieren ..... 3, 4  
Universal-Bereich erzeugen Selektion..... 1  
Viewbereich Objekt ..... 3  
Y-Koordinate der linken oberen Ecke des Objektes..... 2  
Zeiger auf Elternelement des Textbereiches..... 3  
Zeiger auf TextRectangle-Objekt ..... 3

