window.clipboardData Objekt des Internet Explorer

Unterstützung des Zugriffes auf vordefinierte Formate für Clipboard-Nutzung
Es wird das System-Clipboard verwendet. Mehrfacheinfügung ist möglich.
ab IE 5.x
siehe auch Objekt event.dataTransfer des IE

Erzeugung:
durch Browser

Zugriff:

window.clipboardData.methode() oder clipboardData.methode()

logischer_window_name. clipboardData.methode()

logischer_window_name Zeiger laut open()

Beispiel für Clipboard-Nutzung ohne Drag&Drop:

```
<HTML>
<HEAD>
<SCRIPT>
	function Init()
	{
		var TextBereich  = document.body.createTextRange();
		TextBereich.findText(ID_Div1.innerText);
		TextBereich.select();  // selektieren
	}

	function VorDemAusschneiden()
	{event.returnValue = false;} // Cut per Menü aktivieren

	function Ausschneiden()
	{
		ID_Div1.innerText = "";
		ID_Input.innerText += window.clipboardData.setData("Text", ID_Div1.innerText);
					// true oder false
					// Clipboard-Daten sind Texttyp
		event.returnValue = false;
	}

	function VorDemEinfuegen()
	{event.returnValue = false;} // Paste per Menü aktivieren

	function Einfuegen()
	{
		ID_Div2.innerText = window.clipboardData.getData("Text");
		event.returnValue = false;
	}
</SCRIPT>
</HEAD>
<BODY onload="Init()">
	<DIV ID=" ID_Div1" onbeforecut="VorDemAusschneiden()" oncut="Ausschneiden()">
		Dieses Text selektieren und ausschneiden
	</DIV>
	<DIV ID="ID_Div2" onbeforepaste="VorDemEinfuegen()" onpaste="Einfuegen()">
		den ausgeschnittenen Text hier einfügen
	</DIV><BR>
	<INPUT ID="ID_Input" TYPE="text" READONLY VALUE="" SIZE="6">
</BODY>
</HTML>
```

Beispiele für Clipboard-Nutzung mit Drag&Drop:

```
<HEAD>
<SCRIPT>
	function DragStart() // Url eines Ankers als Datenformat festlegen und Daten holen
	{event.dataTransfer.setData("URL", ID_A.href);}

	function DragEnde() // Daten im URL-Format als Textdaten in den Span ablegen
	{ID_Span.innerText = event.dataTransfer.getData("URL");}
</SCRIPT>
</HEAD>
<BODY>
	<A ID="ID_A" HREF="anker"
```

```
                onclick="return(false);"
                ondragstart=" DragStart()"
        >
                Testanker
        </A>
        <SPAN ID="ID_Span" ondragenter=" DragEnde()">
                obigen Link auf diese Stelle hierher droppen
        </SPAN>
</BODY>



<HEAD>
<SCRIPT>
        function InitiateDrag()
        {event.dataTransfer.setData("URL", ID_Image.src);}

        function FinishDrag()
        {ID_Span.innerText = event.dataTransfer.getData("URL");}
</SCRIPT>
</HEAD>
<BODY>
        <IMAGE ID="ID_Image" SRC="/test/graphics/test.gif"
                                        ondragstart="InitiateDrag()">
        <SPAN ID="ID_Span" ondragenter="FinishDrag()"
        >
                Image hierher droppen
        </SPAN>
</BODY>



<HTML>
<HEAD>
<SCRIPT>
        var Wert;

        function TextBereichErzeugen()
        {
                // Text im gesamten Body adressieren
                var TextBereich = document.body.createTextRange();

                // davon den Textteil des Source-Div adressieren
                TextBereich.findText(ID_DivSource.innerText);

                // und diesen selektieren
                TextBereich.select();
        }

        // Ausschneiden aktivieren, da standardgemäß für DIV dekativ ist
        // Ausschneiden bedeutet: Browser verschiebt automatisch in das Clipboard
        function AusschneidenAktivieren()  // ist Eventhandler für onbeforecut
        {event.returnValue = false;}         // Event-Handler-Rückkehrcode

        function Ausschneiden()            // ist Eventhandler für  oncut
        {
                // Clipboard füllen mit Daten vom Texttyp
                //                                      als Text des Source-Div
                Wert = window.clipboardData.setData("Text", ID_DivSource.innerText);

                // Source-Div Textbereich löschen
                ID_DivSource.innerText = "";

                // Rückgabewert vom Clipboardfüllen als Text (true oder false)
                //              im Text des Input-Button anzeigen
                ID_Input.innerText += Wert; // Wert mit als Text

                // Event-Handler-Rückkehrcode
                event.returnValue = false;
        }
        // Einfügen aktivieren, da standardgemäß für DIV dekativ ist
        // Einfügen bedeutet: Browser fügt aus Clipboard in das Zielobjekt ein
        function EinfuegenAktivieren()   // ist Eventhandler für onbeforepaste
        {event.returnValue = false;}       // Event-Handler-Rückkehrcode
```

```
		function Einfuegen()  // ist Eventhandler für onpaste
		{
			// aus Clipboard Daten vom Texttyp holen und in den Textbereich
			// des Target-Div ablegen
			ID_DivTarget.innerText = window.clipboardData.getData("Text");

			// Event-Handler-Rückkehrcode
			event.returnValue = false;
		}
</SCRIPT>
</HEAD>
<BODY	onload="TextBereichErzeugen()">
		<DIV ID="ID_DivSource" onbeforecut="AusschneidenAktivieren()"
					oncut="Ausschneiden()"
		>
			Diesen Text mit Maus markieren und ausschneiden
		</DIV>
		<DIV ID="ID_DivTarget" onbeforepaste="EinfuegenAktivieren()"
					onpaste="Einfuegen()"
		>
			An diese Stelle den ausgeschnittenen Text einfügen
		</DIV>
		<BR>
		<INPUT ID="ID_Input" TYPE="text" VALUE="Rückkehrcode = ">
</BODY>
</HTML>



<HTML>
<HEAD>
<SCRIPT>
		// ###############  Quellobjekt ############################
		function EventHandlerFuerOnDragStart()
			// Quellobjekt löst Event aus:
			//		in Quelle wird markiert und selektiert
		{
			// selektierte Quell-Daten in die Zwischenablage puffern
			//			(Puffer wird per window.event-Objekt verwaltet)
			//		Datentyp ist hier uninteressant, da für die Zwischenablage
			//			der Typ automatisch erkannt wird, also
			//			Speicherung der Daten in der Zwischenablage
			//			immer typgerecht ist
			var ZwischenAblage = window.event.dataTransfer;

			// und diese Daten als verschiebbar erklären:
			//			aus der Quelle in die Zwischenablage
			ZwischenAblage.effectAllowed = "move";
		}

		// ############## Zielobjekt ##############################
		function EventHandlerFuerOnDragEnter
			// Zielobjekt löst Event aus
			// Maus über Ziel nach dem Draggen der Daten,
			//		UND Maustaste ist noch nicht losgelassen
			//   also Zielobjekt wird mit den Daten betreten
		{
			// Daten adressieren
			var ZwischenAblage = window.event.dataTransfer;

			// und diese Daten als verschiebbar erklären:
			//		aus der Zwischenablage in das Zielobjekt
			ZwischenAblage.dropEffect = "move";

			// Event-Handler-Rückkehrcode
			var EventObjekt = window.event;
			EventObjekt.returnValue = false;
		}

		function EventHandlerFuerOnDrop
			// Zielobjekt löst Event aus
			// Maus über Ziel nach dem Droppen der Daten,
```

```
                        //          UND Maustaste wird losgelassen
                {
                        // Ziel adressieren, das mit ondrop-Ereignis behandelt werden soll
                        var Ziel = window.event.srcElement;

                        // Daten in der Zwischenablage adressieren und holen
                        //                      (Puffer wird per window.event-Objekt verwaltet)
                        var ZwischenAblage = window.event.dataTransfer;

                        // und Daten vom Texttyp aus der Zwischenablage im Ziel ablegen,
                        //          da Ziel nur Textdaten empfangen kann
                        Ziel.innerText += Daten.getData("text");

                        // Event-Handler-Rückkehrcode
                        var EventObjekt = window.event;
                        EventObjekt.returnValue = false;
                }

                function EventHandlerFuerOnDragOver
                        // Zielobjekt löst Event aus
                {
                        // nichts tun ausser Rückkehrcode des Handlers liefern
                        var EventObjekt = window.event;
                        EventObjekt.returnValue = false;
                }
        </SCRIPT>
        </HEAD>
        <BODY>
                <DIV ondragstart=" EventHandlerFuerOnDragStart ()"
                >
                        Diesen Text markieren und draggen
                </DIV>
                <BR>
                <DIV    ondragover="EventHandlerFuerOnDragOver()"
                        ondragenter="EventHandlerFuerOnDragEnter()"
                        ondrop="EventHandlerFuerOnDrop()"
                >
                        An diese Stelle den Text droppen
                </DIV>
        </BODY>
        </HTML>
```

Eigenschaften:

keine

Methoden:

.clearData()	Clipboardinhalt löschen Anwendung für Ereignisse ondragstart oder ondrop
.getData()	Clipboard auslesen Anwendung für Ereignisse oncopy oder oncut Handler muss liefern window.event.returnValue=false;
.setData()	Clipboard füllen, also Daten dort ablegen wenn Clipboard nicht leer so immer anhängen