

Format Specification Shapes (.json)

Shape Dateien werden im JSON Format erstellt und unterliegen damit den Anforderungen an dieses Format.

Pro Modell bzw. Serie eines Lieferanten sollte immer eine eigene Shapes Datei erstellt und gepflegt werden.

Shape Dateien bestehen auf der ersten Ebene aus den folgenden Objekten:

- Schema (optional),
- Allgemeine SVG Einstellungen (#svg),
- Allgemeine Möbel Einstellungen (#common) und
- Liste aller Symbole (Shapes).

Für alle Längenwerte gilt: Format ist **Number**, der Dezimaltrenner ist ein **.** und als Einheit wird **m** erwartet.

Diese Dokumentation erfordert eine Version 1.2.5 oder höher des IG.GFX.ShapeCreator.

Schema

In der JSON kann/sollte an erster Stelle die Shapes JSON Schema Datei angegeben werden. Damit können intelligente Editoren wie das Visual Studio Code bereits beim Erstellen der Datei die Gültigkeit der Parameter sowie eine Auto Vervollständigung anbieten.

```
"$schema": "https://archive.intelligentgraphics.biz/schemas/gfx/shapes.json"
```

Allgemeine SVG Einstellungen

In der JSON Datei wird mit Hilfe des Objektes **#svg** dem IG.GFX.ShapeCreator allgemeine Einstellungen zu der SVG Erstellung mitgegeben. Folgende Eigenschaften werden dabei unterstützt:

Name	Typ	Standard	Beschreibung/Werte
Scale	Number	0.1	Die Skalierung für das SVG. Standard ist 0.1.
OffsetX	Int	0	Vergrößert die ViewBox des SVG (Breite) um n Pixel, empfohlen wird 1.
OffsetY	Int	0	Vergrößert die ViewBox des SVG (Höhe) um n Pixel, empfohlen wird 1.
AttachPoints	Boolean	false	Erstellt für alle Symbole Anfügelinien für die Verwendung in XcR-Cat, falls true .
AttachMode	String	LR	Erstellt, falls AttachPoints gleich true, die linke (L) und/oder rechte (R) Anfügelinie. Mit (C) wird ein Modus aktiviert, der zwei Anfügelinien erstellt, welcher in sich öffnenden Ecken sinnvoll sein kann.
AttachOrientation	String	left	Die durch den Modus (AttachMode = "C") erzeugten Anfügelinien werden links am Ursprung ausgerichtet, falls left angegeben, bei right werden diese an der rechten oberen Seite ausgerichtet.

Anfügelinien sollten nur erstellt werden, wenn es sich um planbare Einzelelemente handelt. Bei zusammengesetzten Elementen sollten keine Anfügelinien erstellt werden.

Informationen und Beispiele zu Anfügelinien folgen in einem späteren Abschnitt.

Vererbung

Sämtliche SVG Einstellungen gelten für alle Symbole, sie können aber bei Bedarf pro Symbol überschrieben werden.

Beispiel

```

"#svg": {
  ...
  "AttachPoints": false,
  "AttachMode": "LR"
},
"#common": { ... },
"Shapes": {
  "Armrest_Right": {
    "#svg": {
      "AttachPoints": true,
      "AttachMode": "L"
    },
    "amrest": {
      "#armrest#1": {}
    }
  },
  ...
}

```

Zusätzliche SVG Einstellungen auf Symbolebene

Wie bereits erwähnt, werden sämtliche SVG Einstellungen auf alle untergeordneten Elemente vererbt.

Zusätzlich können aber noch folgende Eigenschaften vergeben werden:

Name	Typ	Standard	Beschreibung/Werte
Product	String		Name des Produkts aus den kaufmännischen Daten, falls der IG.GFX.ShapeCreator eine Liste der Anfügepunkte erstellen soll. Informationen dazu können der Dokumentation des IG.GFX.ShapeCreator entnommen werden.
AttachLeftRotation	Int	0	Die linke Anfügelinie, falls angegeben, wird um 90 oder -90° gedreht.
AttachLeftOrigin	String	origin	Falls origin gesetzt ist, wird die linke Anfügelinie im Ursprung erstellt, falls size angegeben wird, startet die Anfügelinie an der Tiefe des Elementes (gesamte Tiefe).
AttachRightRotation	Int	0	Die rechte Anfügelinie, falls angegeben, wird um 90 oder -90° gedreht.
AttachRightOrigin	String	origin	Falls origin gesetzt ist, wird die rechte Anfügelinie im Ursprung erstellt, falls size angegeben wird, startet die Anfügelinie an der Tiefe des Elementes (gesamte Tiefe).

Allgemeine Möbeleinstellungen

In der JSON Datei wird mit Hilfe des Objektes **#common** dem IG.GFX.ShapeCreator allgemeine Werte für die zu erstellenden Möbel zu der SVG Erstellung mitgegeben. Folgende Eigenschaften werden dabei unterstützt:

Eigenschaft	Typ	Standard	Beschreibung/Werte
Mode	String	cushion	Die Art der Möbel, steuert die Darstellung der Shapes. Aktuell wird nur Polstermöbel (cushion) unterstützt.
ArmrestWidth	Number	0.15	Gibt die Breite der Armlehnen an (Modus cushion).
ArmrestDepth	Number	1.0	Gibt die Tiefe der Armlehnen an (Modus cushion).
BackseatDepth	Number	0.15	Gibt die Tiefe der Rückenkissen an (Modus cushion).
SeatDepth	Number	0.85	Gibt die Sitztiefe der Sofas an (Modus cushion).
Rounding	Number	0.05	Gibt den Standard Durchmesser der gerundeten Ecken an.
Model	String		Name bzw. Id des Modells/Serie für die Erstellung der Symbole.

Hinweise

- ArmrestWidth sollte immer im **#common** Objekt angegeben werden. Ist der Wert nicht bekannt, sollte **0.15** verwendet werden.
- ArmrestWidth wird immer in die Berechnung der Gesamtbreite einbezogen, bei der Bestimmung der Breite einer Couch sollte daher der Wert berücksichtigt werden.
- ArmrestDepth sollte immer im **#common** Objekt angegeben werden. Ist der Wert nicht bekannt, sollte **1.00** verwendet werden.
- BackseatDepth sollte immer im **#common** Objekt angegeben werden. Ist der Wert nicht bekannt, sollte **0.15** verwendet werden.
- SeatDepth sollte immer im **#common** Objekt angegeben werden. Ist der Wert nicht bekannt, sollte **0.85** verwendet werden.
- SeatDepth plus BackseatDepth ergibt die Gesamt Tiefe der Couch.

Beispiel

```
"#common": {
  "Mode": "cushion",
  "ArmrestWidth": 0.15,
  "ArmrestDepth": 0.72,
  "BackseatDepth": 0.15,
  "SeatDepth": 0.57,
  "Rounding": 0.05,
  "Model": "test"
}
```

Symbole

Für jedes Produkt eines Modells bzw. einer Serie, für das ein 2D-Symbol benötigt wird, sollte im Objekt Shapes ein Symbol definiert werden. Der Name des Objektes gibt dabei auch den Dateinamen vor, es sollten daher sprechende Produktnamen verwendet werden. Auf Zeichen außerhalb des Alphabets und Nummern sowie _ oder - sollte dabei verzichtet werden. Leerzeichen sind auf jeden Fall zu vermeiden.

Jedes Symbol wiederum besteht aus einer Liste von **Bauteilen**, die wie ein Bauplan zu verstehen sind. Jedes Bauteil wiederum kann aus einer Reihe von definierten **Zeichenelementen** bestehen.

Bauteile und Zeichenelemente wiederum können definierte Eigenschaften besitzen, die die Erzeugung der SVG Symbole beeinflussen.

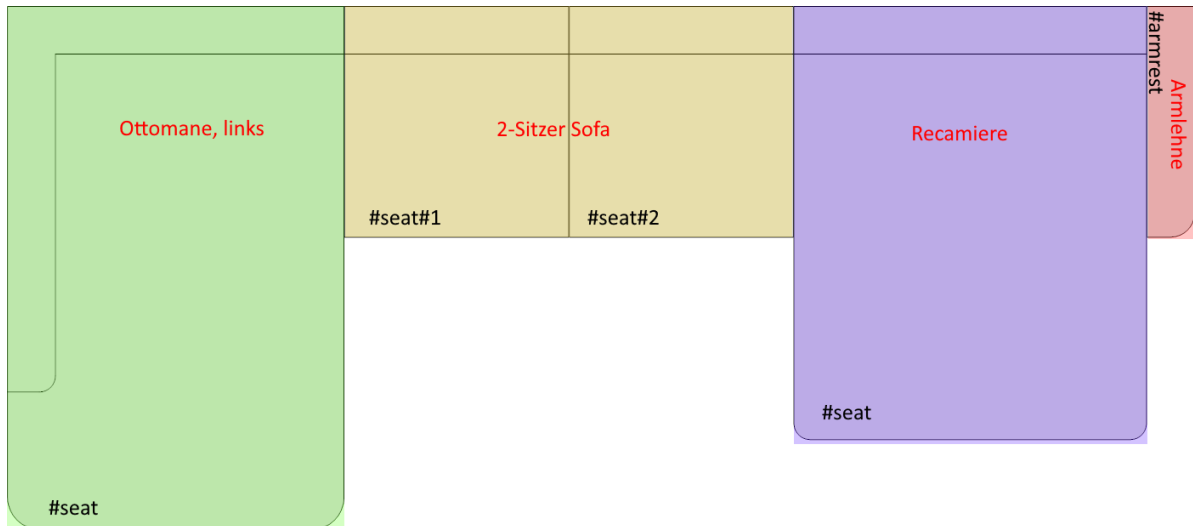


Abbildung: Bauteile (roter Text) und zugehörige Zeichenelemente (schwarzer Text)

Am besten lässt sich das an den folgenden Beispielen nachvollziehen.

Beispiel 1, Hocker

```
"Hocker_100x100": {
  "hocker": {
    "width": 1.0,
    "Depth": 1.0,
    "#box": {
      "corner": "round"
    }
  }
}
```

Ein Hocker besteht in der Regel aus genau einem Bauteil, hier **"hocker"**. Als Breite und Tiefe werden dabei 1.0 m angegeben. Zur Darstellung wird das definierte Zeichenelement **#box** verwendet, hier mit runden Recken. Der Durchmesser der Ecken wird aus den allgemeinen Möbeleinstellungen (`#common`, `Rounding`) entnommen.

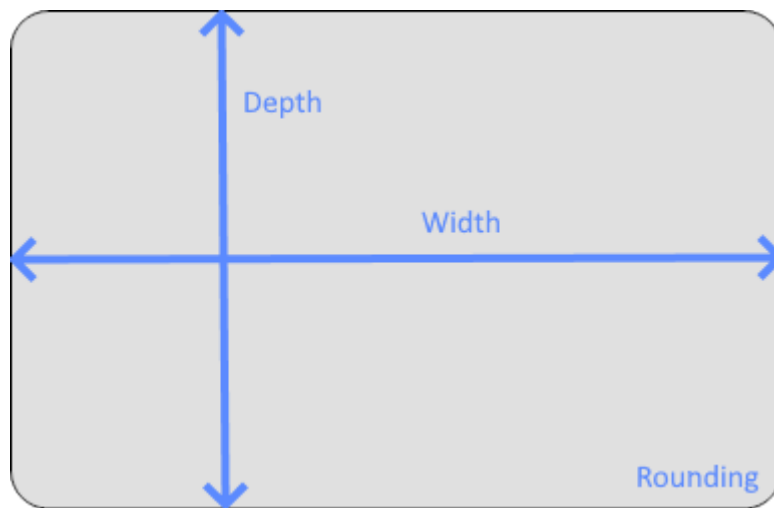


Abbildung: Hocker mit Zeichenelement #box und den Eigenschaften

Width und **Depth** sind Eigenschaften des Bauteil "**hocker**", corner ist eine Eigenschaft des definierten Zeichenelementes #**box**.

Beispiel 2, Kombination aus 2-Sitzer und Recamiere

```
"2SEATER_REC_RECHTS": {
  "2": {
    "width": 1.4,
    "#armrest#1": {
      "corner_3": "round",
      "diameter_3": 0.075
    },
    "#seat#1": {},
    "#seat#2": {}
  },
  "rec": {
    "width": 1.1,
    "#seat#1": {
      "SeatDepth": 1.2,
      "corner_2": "round",
      "diameter_2": 0.05,
      "corner_3": "round",
      "diameter_3": 0.05
    },
    "#armrest#1": {
      "corner_2": "round",
      "diameter_2": 0.075
    }
  }
}
```

Diese Kombination hat eine Gesamtbreite von 2.8 m sowie eine Gesamttiefe von 1.35 m (Recamiere). Das Sofa selbst ist 0.72 m tief bei einer Rückenteiltiefe von 0.15 m.

Die Kombination besteht aus den Bauteilen "2" (2-Sitzer Sofa mit Armlehne links) und "rec" (Recamiere mit Armlehne rechts), wobei das Sofa gesamt 1.55 m breit ist (inkl. Armlehne) und die Recamiere gesamt 1.25 m breit ist (inkl. Armlehne).

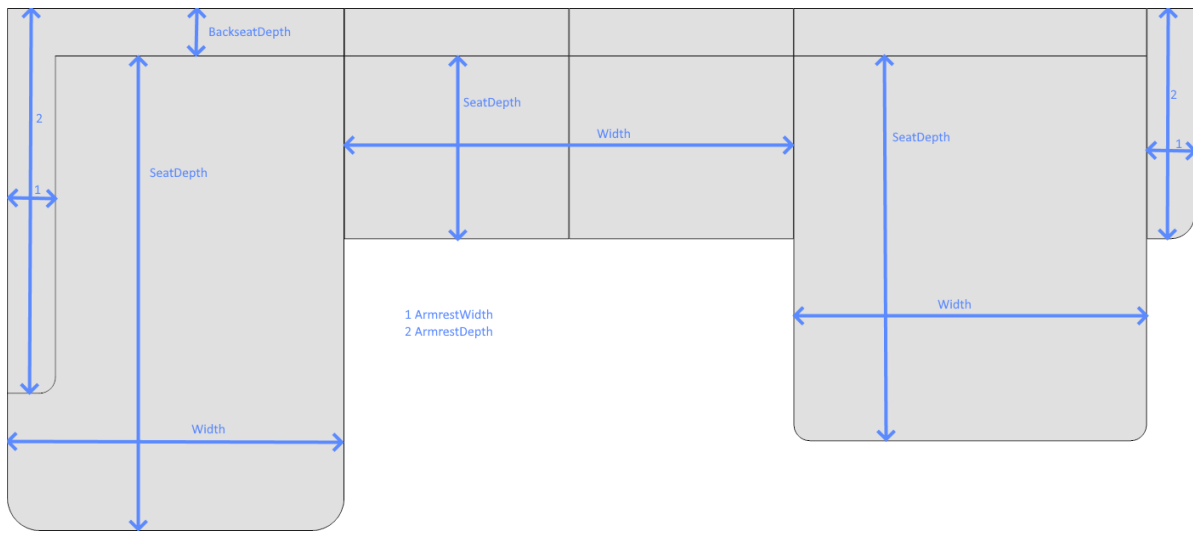


Abbildung: Kombination von mehreren Bauteilen und Elementen und den Eigenschaften (exemplarisch)

Für die Darstellung des Sofas und der Recamiere werden die Zeichenelemente **#armrest** und **#seat** verwendet. Je nach Bedarf werden diese über unterstützte Eigenschaften angepasst.

Beispiel 3, Kombination mit Ecke

Das folgende Beispiel zeigt ein Eckkombination.

```
"2_e_3": {
  "2": {
    "width": 1.4,
    "Rotation": -90,
    "#armrest#1": {
      "corner_3": "round",
      "diameter_3": 0.075
    },
    "#seat#1": {},
    "#seat#2": {}
  },
  "e": {
    "#corner#1": {
      "width": 0.83,
      "Depth": 0.83,
      "Orientation": "left"
    }
  },
  "3": {
    "width": 2.1,
    "#seat#1": {},
    "#seat#2": {},
    "#seat#13": {},
    "#armrest#1": {
      "corner_2": "round",
      "diameter_2": 0.075
    }
  }
}
```

}
}

Bei Ecken ist es wichtig, die entsprechenden Bauteile vor Ecke zu drehen mit Hilfe der Eigenschaft **Rotation**. In diesem Fall ist es links ausgerichtete Ecke mit einer Rotation von -90° , für rechts ausgerichtete Ecken muss die Rotation auf 90° gesetzt werden.

Eigenschaften von Bauteilen

Bauteile unterstützen die folgenden Eigenschaften:

Eigenschaft	Typ	Standard	Beschreibung/Werte
Width	Number	1.0	Gibt die Gesamtbreite des Bauteils an.
Depth	Number	1.0	Gibt die Gesamttiefe des Bauteils an.
Rotation	Grad	0	Rotiert das gesamte Bauteil (inkl. der Elemente) um n Grad.
Orientation	String	none	Richtet das Bauteil aus. falls unterstützt. Mögliche Werte sind none , left oder right .
ArmrestWidth	Number	0.15	Gibt die Breite der Armlehnen an (Modus cushion).
ArmrestDepth	Number	1.0	Gibt die Tiefe der Armlehnen an (Modus cushion).
BackseatDepth	Number	0.15	Gibt die Tiefe der Rückenkissen an (Modus cushion).
SeatDepth	Number	0.85	Gibt die Sitztiefe der Sofas an (Modus cushion).

Vererbung

Sämtliche Bauteile und Zeichenelemente verwenden die in den allgemeinen Einstellungen `#common` definierten Werte. Diese können aber bei Bedarf, wie in der Tabelle oben angegeben, pro Bauteil (oder Zeichenelement) überschrieben werden.

Eigenschaften von Zeichenelementen

Folgende Zeichenelemente sind im Modus "**cushion**" definiert und können in den Bauteilen verwendet werden:

- `#box`
Eine Box mit Width und Height sowie optionalen runden Ecken
- `#seat`
Ein Sitz mit Rückenlehne
Durch die Eigenschaft "**RenderMode**" kann die Darstellung für verschiedene Verwendungszwecke angepasst werden, siehe Übersicht weiter unten
- `#armrest`
Eine Armlehne mit optional gerundeten Ecken, siehe Übersicht unten
- `#corner`
Ecken sind spezielle Zeichenelemente, um Kombinationen um Ecken planen zu können

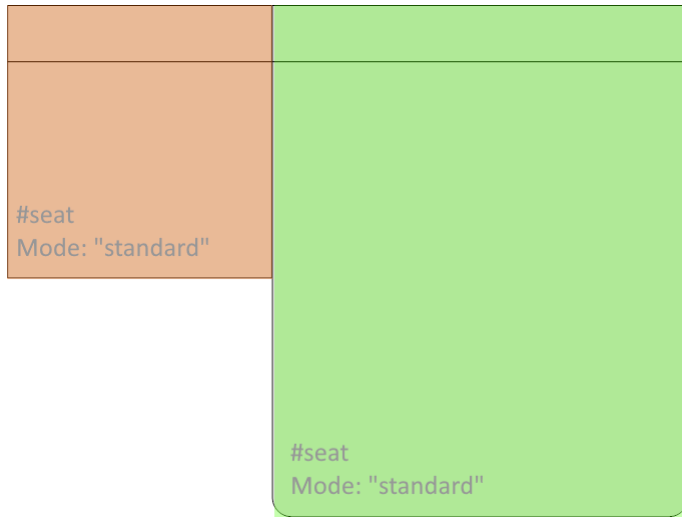
Jedes Zeichenelement kann folgende Eigenschaften besitzen:

Name	Typ	Standard	Beschreibung/Werte
Width	Number	1.0	Breite für das Zeichenelement setzen.
Depth	Number	1.0	Tiefe für das Zeichenelement setzen.
ArmrestWidth	Number	0.15	Gibt die Breite der Armlehnen an (Modus cushion).
ArmrestDepth	Number	1.0	Gibt die Tiefe der Armlehnen an (Modus cushion).
BackseatDepth	Number	0.15	Gibt die Tiefe der Rückenkissen an (Modus cushion).
SeatDepth	Number	0.85	Gibt die Sitztiefe der Sofas an (Modus cushion).
Orientation	String	none	Richtet das Zeichenelement aus, falls unterstützt. Mögliche Werte sind none , left oder right .
RenderMode	String	standard	Ändert die Darstellung des(r) erzeugten Sitzes, Ecke und oder anderer Grundtypen.
corner	String	corner	Ändert die Darstellung aller Ecken, unterstützt wird corner (Ecken) oder round (Gerundete Ecken). Nur für Zeichenelement #box !
diameter	Number	0.05	Gibt den Durchmesser für die gerundeten Ecken an. Nur für Zeichenelement #box !
corner_2	String	corner	Falls "round" wird die <u>rechte</u> untere Ecke des Sitzes gerundet. Nur für Zeichenelemente #seat und #armrest !
diameter_2	Number	0.05	Gibt den Durchmesser für die gerundete Ecke an.
corner_3	String	corner	Falls "round" wird die <u>linke</u> untere Ecke des Sitzes gerundet. Nur für Zeichenelemente #seat und #armrest !
diameter_3	Number	0.05	Gibt den Durchmesser für die gerundete Ecke an.

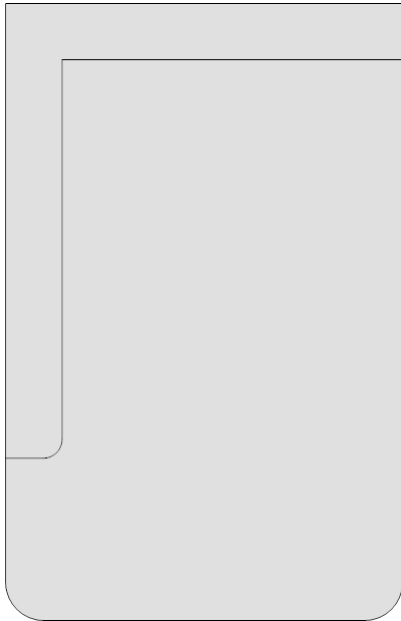
RenderMode (Zeichenelement #seat)

Folgende Darstellungen kann das Element #seat annehmen:

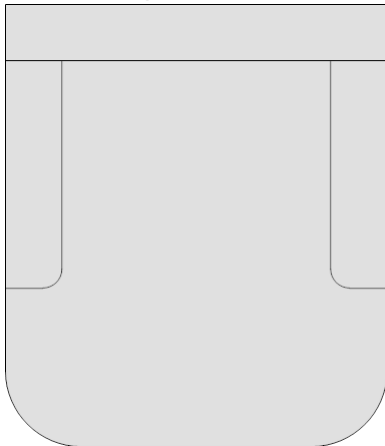
- Modus "standard", Normaler Sitz (z.B. n-Sitzer oder Recamiere)



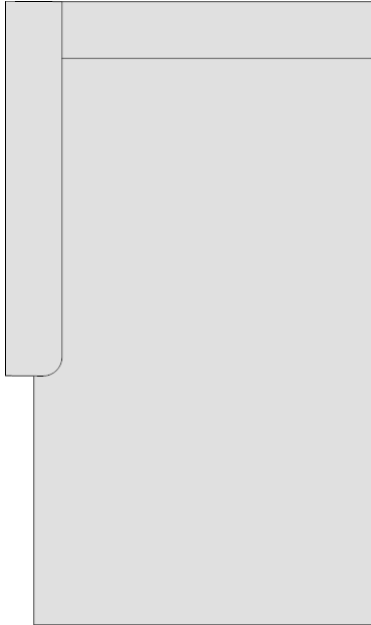
- Modus "ottomane", für Ottomanen



- Modus "single", für Sessel



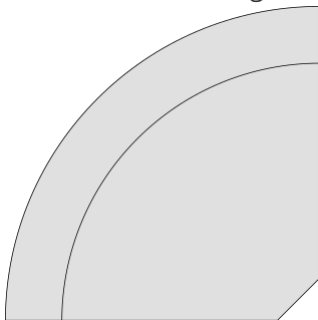
- Modus "divan", für Divans



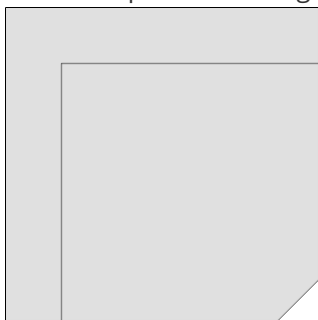
RenderMode (Zeichenelement #corner)

Folgende Darstellungen kann das Element #corner annehmen:

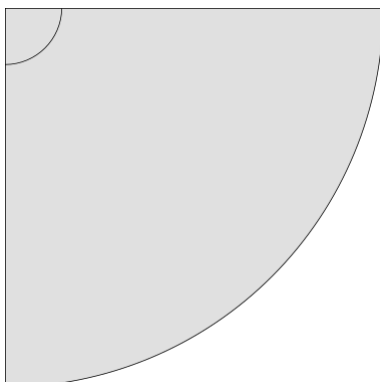
- Standard-Darstellung



- Modus "square" für eckige Ecken



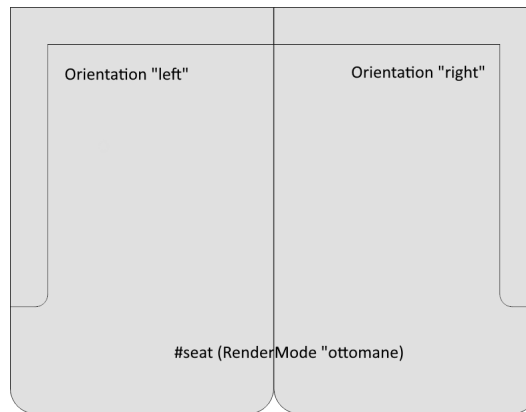
- Modus "roundMirror" für 270° Ecken, runde Darstellung



Ausrichtung (Orientation)

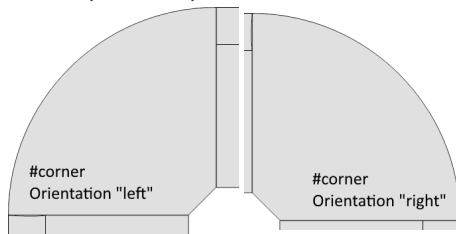
Folgende Zeichenelement unterstützen verschiedene Ausrichtungen:

- Ottomanen (#seat mit RenderMode "ottomane")



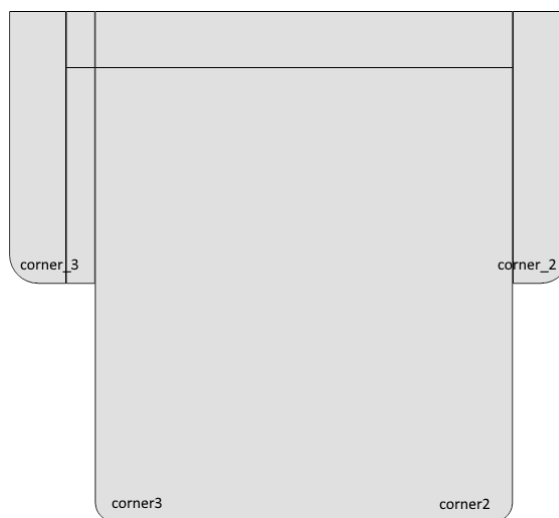
Ottomanen mit der Ausrichtung "links" sind Abschlusselemente auf der linken Seite und vice versa.

- Ecken (#corner)



Rundungen von Ecken (Zeichenelemente #seat und #armrest)

Sitze und Armlehnen unterstützen die Abrundung der unter linken (3) und rechten Ecke (2).



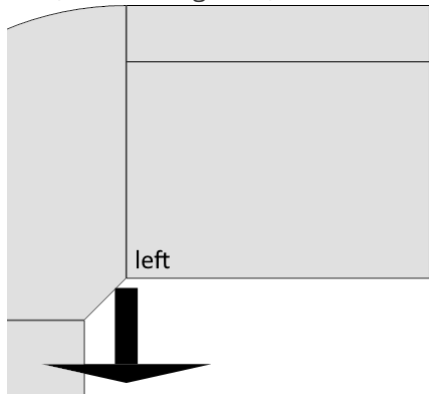
Erweiterungen

Zeichenelemente können verschiedene Erweiterungen erhalten, die zur Darstellung von weiteren Funktionen verwendet werden können. Dazu zählen z.B. eine Ausziehfunktion, LED Beleuchtungen oder elektronische Sitztiefen Verstellungen.

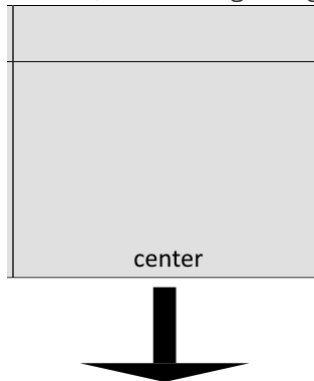
PullOut

Ein PullOut Element stellt einen Pfeil zur Darstellung einer Ausziehfunktion dar. Mit Hilfe der Eigenschaft "Align" kann der Pfeil am zugehörigen Zeichenelement ausgerichtet werden. Dabei werden:

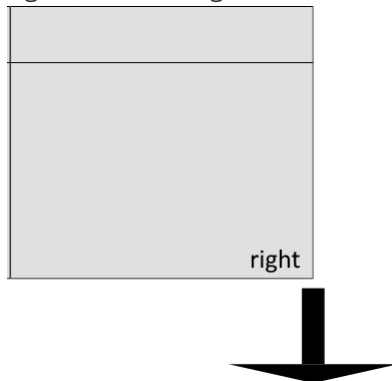
- left (Ausrichtung links),



- center (Ausrichtung mittig) und



- right (Ausrichtung rechts)



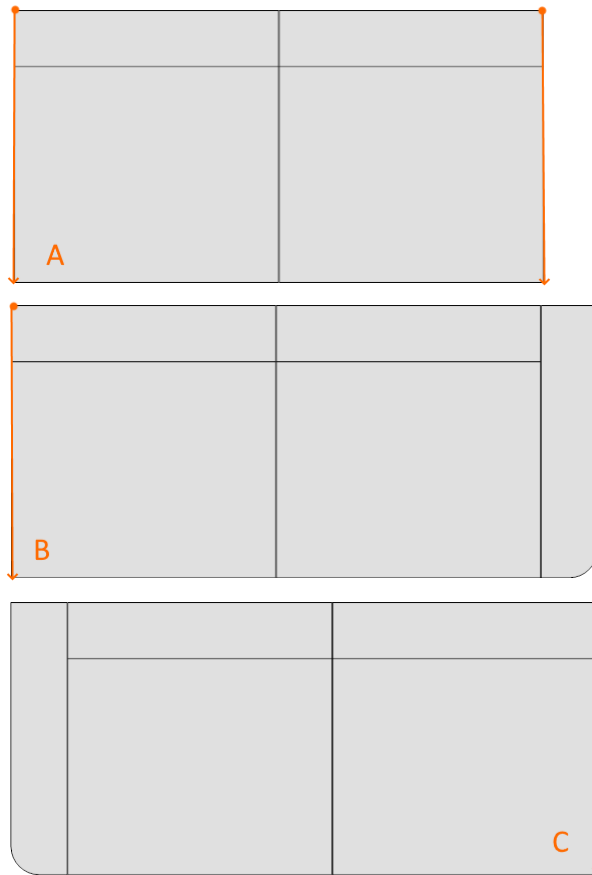
Anfügelinien im #svg Element

Die Eigenschaft "**AttachMode**" gibt dem IG.GFX.ShapeCreator vor, welche Anfügelinien erzeugt werden.

Beispiel

```
"#svg": {  
  "AttachPoints": true,  
  "AttachMode": "LR",  
}
```

Die folgende Grafik zeigt dabei die Standard Modi "LR" (A), "L" (B) und "R" (C).



Ursprung

Der Ursprung der Anfügelinien im Modus "L" und "R" kann verschoben werden. Dazu können die Eigenschaften "**AttachLeftOrigin**" und "**AttachRightOrigin**" verwendet werden und auf den Wert size gesetzt werden. Der Standardwert ist origin, dieser muss aber nicht explizit angegeben werden.

Diese Verschiebung ist z.B. bei ECKELEMENTen sinnvoll, um den richtigen Anfügepunkt definieren zu können.

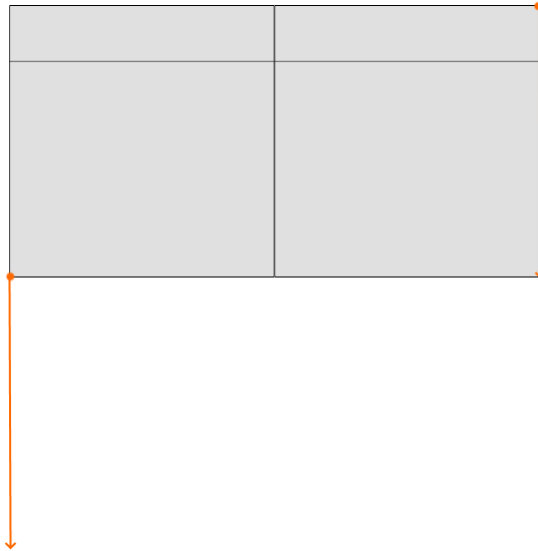
Beispiel

```

"#svg": {
  "AttachPoints": true,
  "AttachMode": "LR",
  "AttachLeftOrigin": "size"
}

```

Die folgende Grafik zeigt den verschobenen Ursprung der Anfügelinien, wenn Origin mit "size" angegeben wird.



Hinweis: Die Länge der Linien ergibt sich immer aus der gesamten Tiefe des erzeugten Shapes.

Rotation

Unter Umständen ist es notwendig, dass die Anfügelinien gedreht werden müssen, wenn Elemente in andere Richtung angefügt werden müssen. Dazu können die Eigenschaften "**AttachLeftRotation**" und "**AttachRightRotation**" verwendet werden.

Die folgenden Beispiele und die zugehörige Abbildung zeigen die möglichen Werte für die Rotation.

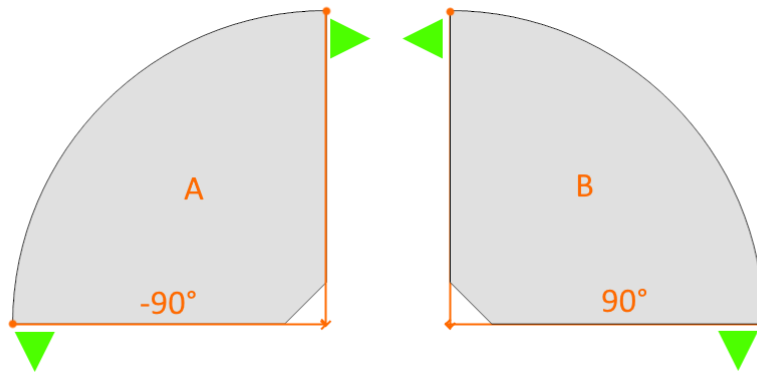
Variante **A**, die linke Anfügelinie wird im Ursprung verschoben und dann gedreht, um nach "rechts" zu zeigen.

```
"#svg": {  
  "AttachPoints": true,  
  "AttachMode": "LR",  
  "AttachLeftRotation": -90,  
  "AttachLeftOrigin": "size"  
}
```

Variante **B**, die rechte Anfügelinie wird im Ursprung verschoben und dann gedreht, um nach "links" zu zeigen.

```
"#svg": {  
  "AttachPoints": true,  
  "AttachMode": "LR",  
  "AttachRightRotation": 90,  
  "AttachRightOrigin": "size"  
}
```

Die Abbildung zeigt die Ausrichtung und Drehung der Anfügelinien sowie die gedachten Anfügerichtungen.



Öffnende Ecken

Unter Umständen können Ecken auch einen Winkel von 270° besitzen, diesen Umstand unterstützt der IG.GFX.ShapeCreator mit dem Wert "C" im Merkmal "**AttachMode**". In diesem Fall werden zwei passende Anfügelinien automatisch erzeugt und müssen nicht zusätzlich gedreht werden. Beide Anfügelinien haben einen gemeinsamen Ursprung. Dieser kann mit der Eigenschaft "**AttachOrientation**" an die linke (Wert "left", Standard) oder rechte obere Ecke (Wert "right") gesetzt werden.

Im Beispiel A wird eine linke Ausrichtung, im Beispiel B eine rechte Ausrichtung verwendet.

Hinweis: im Modus "C" kann der Ursprung **nicht** verschoben werden wie in den Modi "L", "R" oder "LR".

Beispiel A

```
"#svg": {
  "AttachPoints": true,
  "AttachMode": "C",
  "AttachOrientation": "left"
}
```

Beispiel B

```
"#svg": {
  "AttachPoints": true,
  "AttachMode": "C",
  "AttachOrientation": "right"
}
```

Das folgende Bild zeigt das Resultat der Eigenschaften im #svg Objekt.

