

Übung 4: Text und Tastatur

Diese Übung dreht sich um den Umgang mit Tastatureingaben und die Darstellung von Text.

Aufgabe 1: Tastatureingaben

Processing bietet die folgenden Elemente für das Verarbeiten von Tastatureingaben:

Element	Beschreibung
<code>keyPressed()</code>	Methode. Wird aufgerufen, wenn eine Taste gedrückt wird.
<code>keyReleased()</code>	Methode. Wird aufgerufen, wenn eine Taste losgelassen wird.
<code>keyPressed</code>	Variable. Gibt an ob eine Taste gedrückt wird (<code>boolean</code>).
<code>key</code>	Variable. Enthält Wert der zuletzt gedrückten Taste als <code>char</code> oder <code>CODED</code> , falls es sich um eine Spezial-Taste handelt (s. u.).
<code>keyCode</code>	Variable. Enthält Wert der zuletzt gedrückten Spezial-Taste als Konstante (z.B. <code>UP</code> , <code>DOWN</code> , <code>LEFT</code> , <code>RIGHT</code> , <code>ALT</code> , <code>CONTROL</code> , <code>SHIFT</code>)
<code>CODED</code>	Konstante. Dient zur Feststellung, ob eine gedrückte Taste eine Spezial-Taste ist: <code>if (key == CODED) {...}</code>

→ Erstellen Sie einen einfachen Sketch, der bei Tastendruck den entsprechenden (Klein-) Buchstaben auf der Konsole ausgibt. Eine grafische Ausgabe ist noch nicht erforderlich.

→ Verändern Sie das Programm nun so, dass nicht jedes Zeichen einzeln, sondern immer der gesamte bisherige Text auf der Konsole ausgegeben wird.

Versuchen Sie nun auch Großbuchstaben einzutippen. Welches Problem ergibt sich dabei offenbar?

→ Erweitern Sie das Programm um die Funktionalität für Großbuchstaben. Dazu müssen Sie lediglich sicherstellen, dass kein Wert gespeichert/ausgegeben wird, wenn es sich bei der gedrückten Taste um eine Spezial-Taste handelt.

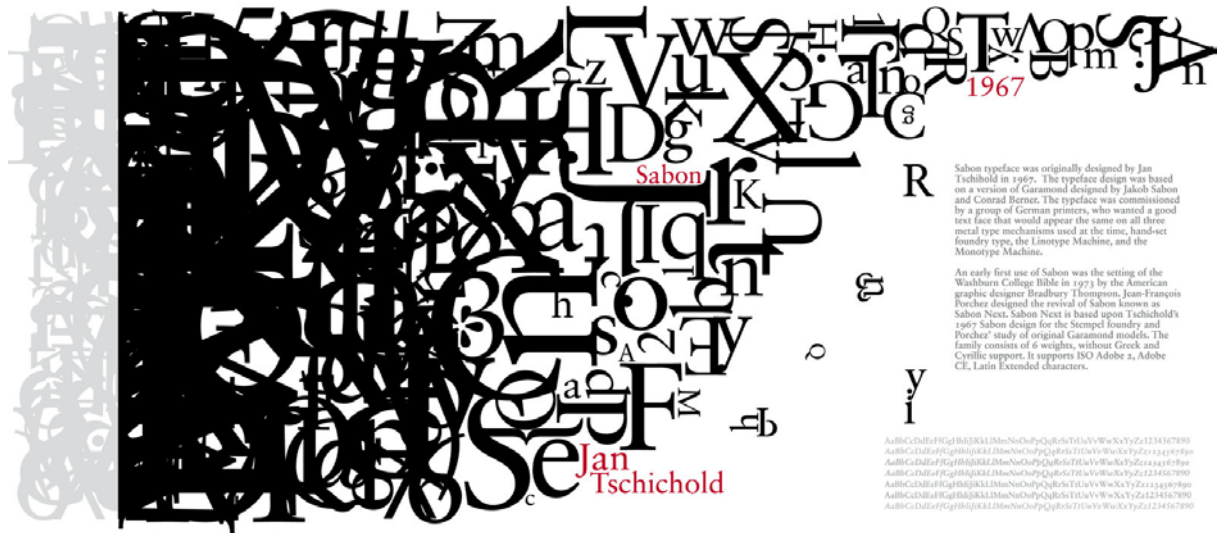
→ Erweitern das Programm so, dass durch Druck der Backspace-Taste das letzte Zeichen wieder entfernt wird. Die Backspace-Taste hat den Wert `BACKSPACE` und ist *keine* Spezial-Taste. Tipp: Für das Kürzen eines Strings eignet sich die Methode `substring(startIndex, endIndex)`.

Aufgabe 2: Textausgabe

Zur grafischen Ausgabe von Text mit Processing dienen die folgenden Funktionen:

Funktion	Beschreibung
<code>loadFont („fontname.vlw“)</code>	Lädt einen Font aus der angegebenen Datei und gibt ein entsprechendes <code>PFont</code> -Objekt zurück.
<code>textFont(myFont, size)</code>	Setzt die Schrift auf das angegebene <code>PFont</code> -Objekt / Größe.
<code>text („text“, x, y)</code>	Gibt den Text an der angegebenen Position aus.

- Erweitern Sie das Programm aus Aufgabe 1 um eine grafische Darstellung des Textes. Font-Dateien können in der Processing-IDE über den Menüpunkt „Tools → Font...“ erzeugt werden.
- Optional: Experimentieren sie mit Schriftgröße, Farbe und Textposition. Versuchen sie Text zufällig erzeugen zu lassen. Zur Inspiration:



<http://dougalops.deviantart.com/art/Sabon-typography-poster-186262380>