

```

// Settings
// =====
$hamburger-layer-width      : 30px !default;
$hamburger-layer-height    : 3px !default;
$hamburger-layer-spacing    : 6px !default;
$hamburger-layer-color      : #444 !default;
$hamburger-layer-border-radius : 10px !default;
$hamburger-hover-opacity    : 0.7 !default;
$hamburger-hover-transition-duration : 0.9s !default;
$hamburger-hover-transition-timing-function: linear !default;

// To use CSS filters as the hover effect instead of opacity,
// set $hamburger-hover-use-filter as true and
// change the value of $hamburger-hover-filter accordingly.
$hamburger-hover-use-filter: false !default;
$hamburger-hover-filter : opacity(50%) !default;

/* hamburger */
/*
 * Elastic
 */
.hamburger {
    Display: inline-block;
    Cursor: pointer;
    Transition-property: opacity, filter;

```

Transition-duration: \$hamburger-hover-transition-duration;

Transition-timing-function: \$hamburger-hover-transition-timing-function;

// Normalize (<button>)

Font: inherit;

Color: inherit;

Text-transform: none;

Background-color: transparent;

Border: 0;

Margin: 0;

Overflow: visible;

&:hover {

 @if \$hamburger-hover-use-filter == true {

 Filter: \$hamburger-hover-filter;

 }

 @else {

 Opacity: \$hamburger-hover-opacity;

 }

}

}

.hamburger-box {

 Width: \$hamburger-layer-width;

 Height: \$hamburger-layer-height * 3 + \$hamburger-layer-spacing * 2;

 Display: inline-block;

 Position: relative;

```
}
```

```
.hamburger-inner {
```

```
    Display: block;
```

```
    Top: 50%;
```

```
    Margin-top: $hamburger-layer-height / -2;
```

```
    &,
```

```
    &::before,
```

```
    &::after {
```

```
        Width: $hamburger-layer-width;
```

```
        Height: $hamburger-layer-height;
```

```
        Background-color: $hamburger-layer-color;
```

```
        Border-radius: $hamburger-layer-border-radius;
```

```
        Position: absolute;
```

```
        Transition-property: transform;
```

```
        Transition-duration: 0.3s;
```

```
        Transition-timing-function: ease;
```

```
    }
```

```
    &::before,
```

```
    &::after {
```

```
        Content: "";
```

```
        Display: block;
```

```
    }
```

```
    &::before {
```

```

        Top: ($hamburger-layer-spacing + $hamburger-layer-height) * -1;
    }

    &::after {
        Bottom: ($hamburger-layer-spacing + $hamburger-layer-height) * -1;
    }
}

.hamburger—elastic {
    .hamburger-inner {
        Top: $hamburger-layer-height / 2;
        Transition-duration: 0.45s;
        Transition-timing-function: cubic-bezier(0.68, -0.55, 0.265, 1.55);

        &::before {
            Top: $hamburger-layer-height + $hamburger-layer-spacing;
            Transition: opacity 0.43s 0.275s ease;
        }

        &::after {
            Top: ($hamburger-layer-height * 2) + ($hamburger-layer-spacing * 2);
            Transition: transform 0.4s cubic-bezier(0.68, -0.55, 0.265, 1.55);
        }
    }
}

&.is-active {
    .hamburger-inner {

```

\$y-offset: \$hamburger-layer-spacing + \$hamburger-layer-height;

Transform: translate3d(0, \$y-offset, 0) rotate(135deg);

Transition-delay: 0.075s;

&::before {

Transition-delay: 0s;

Opacity: 0;

}

&::after {

Transform: translate3d(0, \$y-offset * -2, 0) rotate(-270deg);

Transition-delay: 0.075s;

}

}

}

}