```
(function(){
  var obj = {
      winW: window.innerWidth-17,
      winH: window.innerHeight-17,

      init:function(){
          this.winFn();
          this.scr();
          console.log('winW : ', this.winW);
          console.log('winH : ', this.winH);
          console.log('winH .section3OffsetTop : ', this.scr.section3OffsetTop);
          console.log('winH .section4OffsetTop : ', this.scr.section4OffsetTop);
          console.log('winH .section5OffsetTop : ', this.scr.section5OffsetTop);
          console.log('winH .section6OffsetTop : ', this.scr.section6OffsetTop);
      },
      scr: function(){
          // var section3 = document.getElementById('#section3').getBoundingClientRect().top;
          //        console.log('section3.getBoundingClientRect().top ', section3 );




      },
      winFn: function(){
        var screenW = screen.width;
            console.log( 'screenW ', screenW );
        var availWidth = screen.availWidth;
            console.log( 'availWidth ', availWidth );
        var winWin = window.innerWidth;
        var winHin = window.innerHeight;
            console.log( 'innerWidth 창너비는 ', winWin );
            console.log( 'innerHeight 창높이는 ', winHin );
            console.log( 'innerHeight-17 창높이는 ', winHin-17 );
            console.log( 'innerWidth-17 내부 창너비는 ', winWin-17 );
        var winWout = window.outerWidth;
            console.log( 'outerWidth 창너비는 ', winWout );
        var wrap = document.getElementById('wrap').offsetWidth;
            console.log( 'offsetWidth wrap 너비 ', wrap );
        var winW2 = document.body.offsetWidth;
            console.log( 'offsetWidth 창너비는 ', winW2 );
        var winW2H = document.body.offsetHeight;
            console.log( 'offsetHeight 창높이는 ', winW2H );
      }
  }

  obj.init();

})();
```

https://developer.mozilla.org/en-US/docs/Web/API/Element/getBoundingClientRect

Element.getBoundingClientRect()

```
domRect = element.getBoundingClientRect();
```

자바 스크립트

JavaScript 코드는 CSS 클래스가 withClientRectsOverlay할당 된 모든 HTML 요소에 대한 클라이언트 rects를 그립니다 .

```
function addClientRectsOverlay(elt) {
  /* Absolutely position a div over each client rect so that its border width
     is the same as the rectangle's width.
     Note: the overlays will be out of place if the user resizes or zooms. */
  var rects = elt.getClientRects();
  for (var i = 0; i != rects.length; i++) {
    var rect = rects[i];
    var tableRectDiv = document.createElement('div');
    tableRectDiv.style.position = 'absolute';
    tableRectDiv.style.border = '1px solid red';
    var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;
    var scrollLeft = document.documentElement.scrollLeft || document.body.scrollLeft;
    tableRectDiv.style.margin = tableRectDiv.style.padding = '0';
    tableRectDiv.style.top = (rect.top + scrollTop) + 'px';
    tableRectDiv.style.left = (rect.left + scrollLeft) + 'px';
    // We want rect.width to be the border width, so content width is 2px less.
    tableRectDiv.style.width = (rect.width - 2) + 'px';
    tableRectDiv.style.height = (rect.height - 2) + 'px';
    document.body.appendChild(tableRectDiv);
  }
}

(function() {
  /* Call function addClientRectsOverlay(elt) for all elements with
     assigned class "withClientRectsOverlay" */
  var elt = document.getElementsByClassName('withClientRectsOverlay');
  for (var i = 0; i < elt.length; i++) {
    addClientRectsOverlay(elt[i]);
  }
})();
```

결과