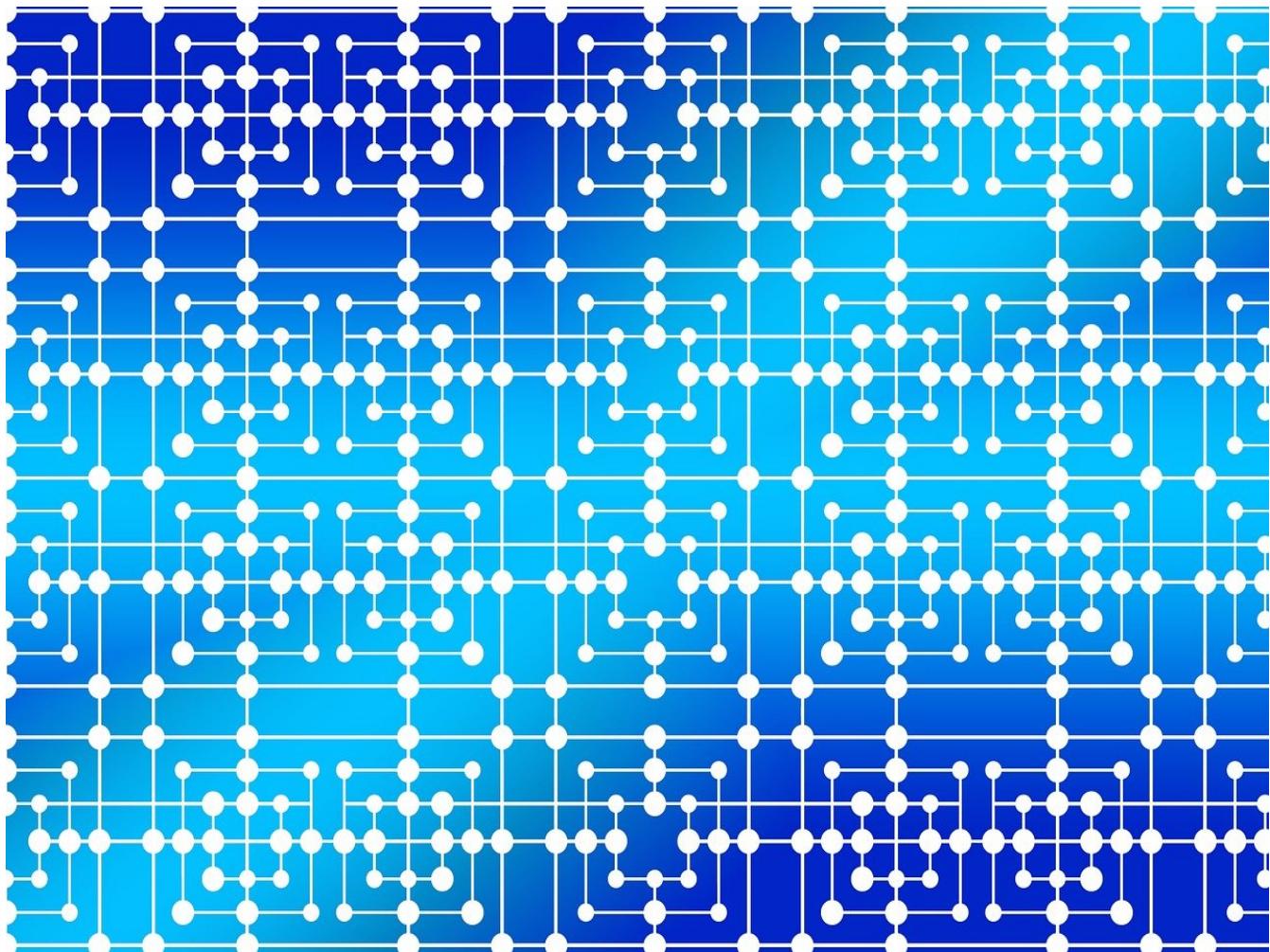
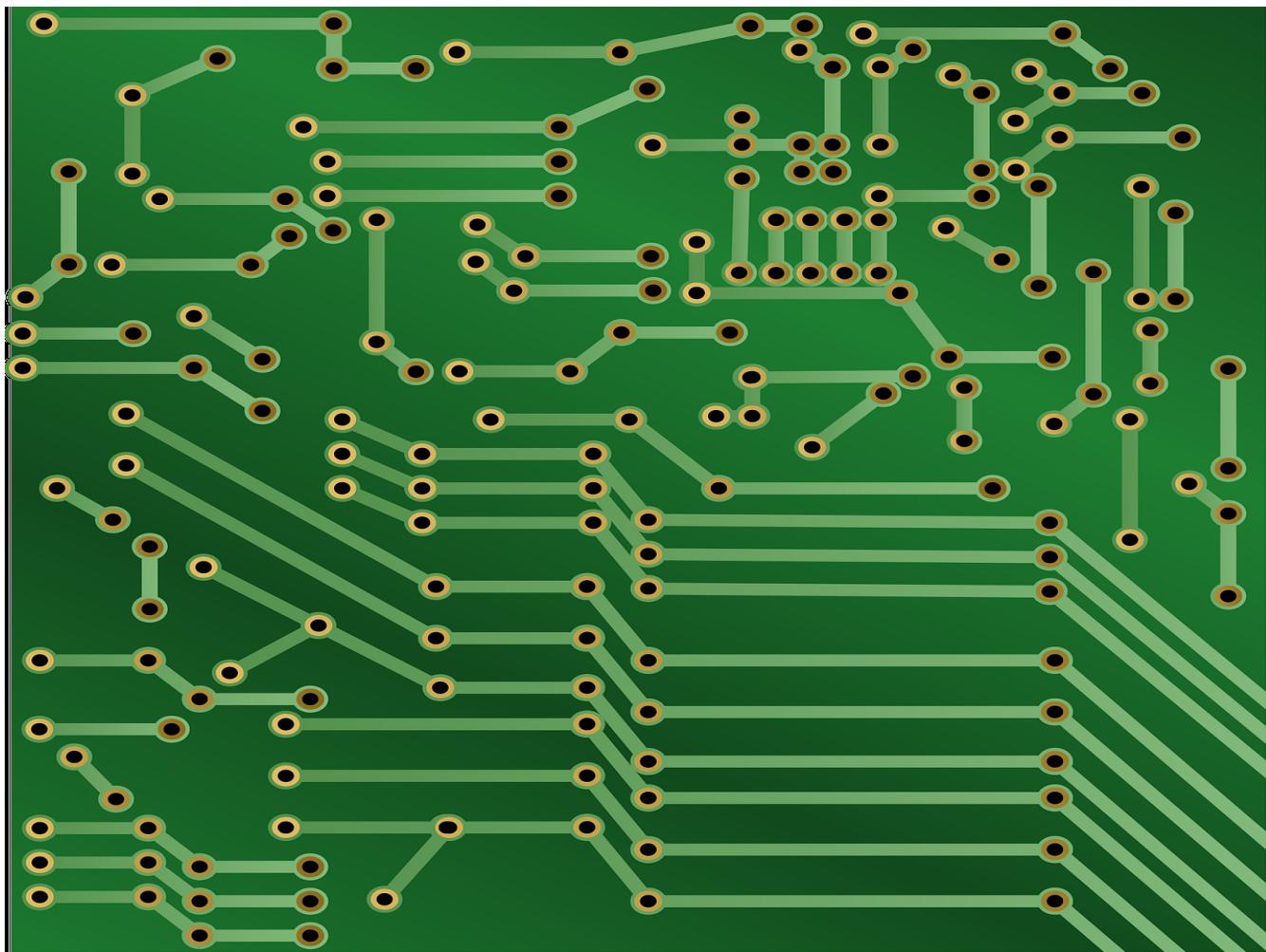


Reference 1



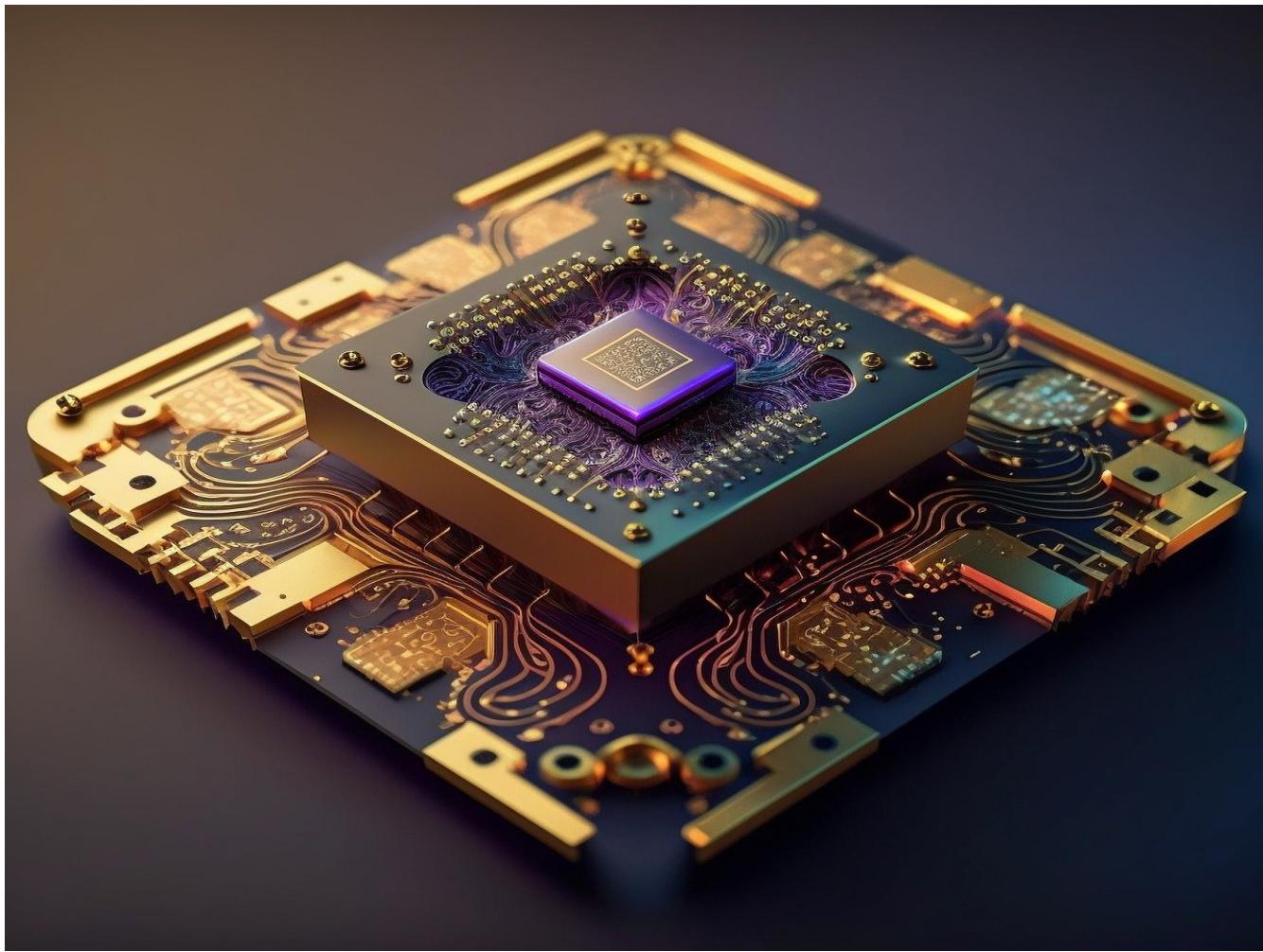
Reference 2



Reference 3



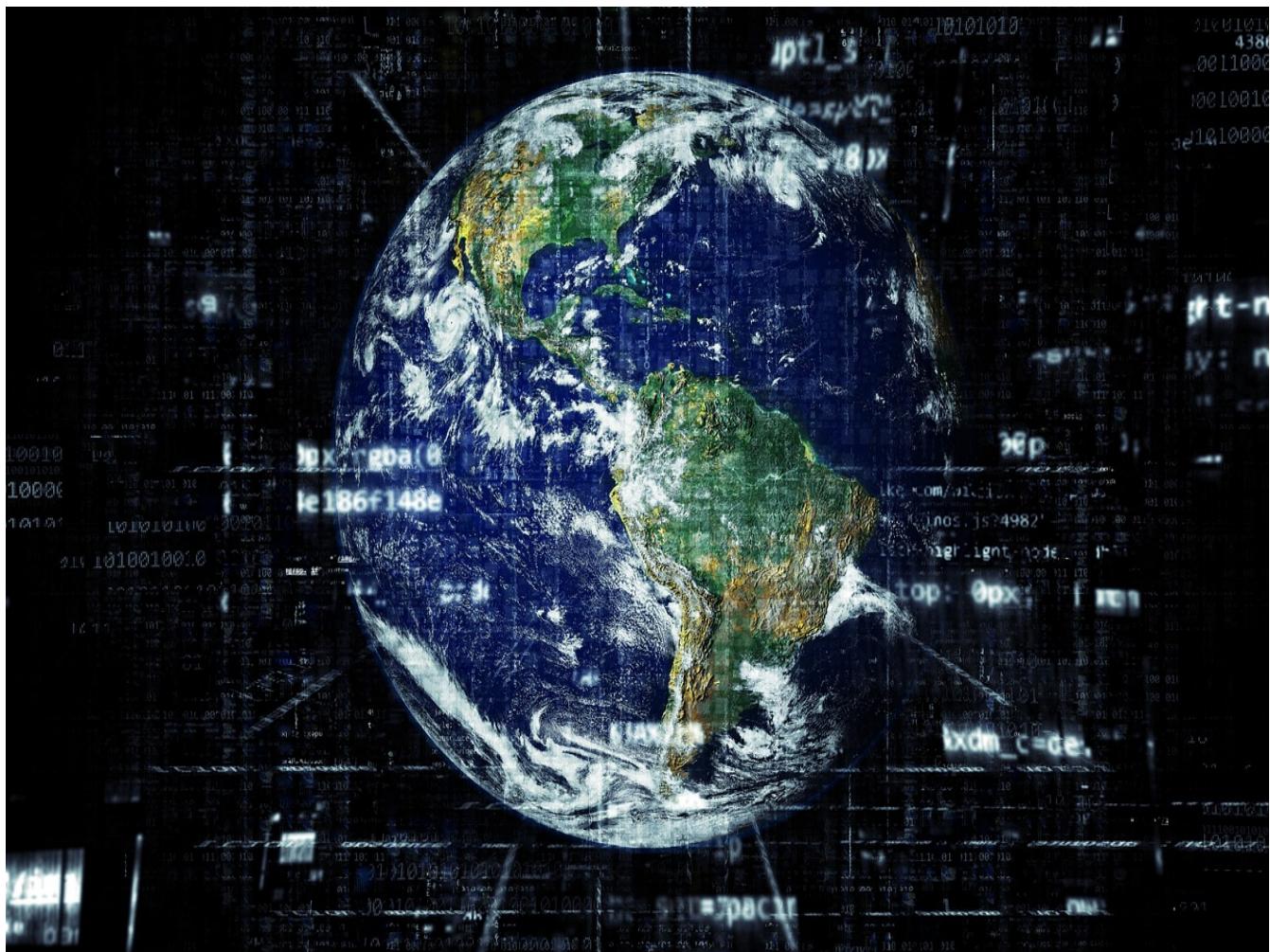
Reference 4



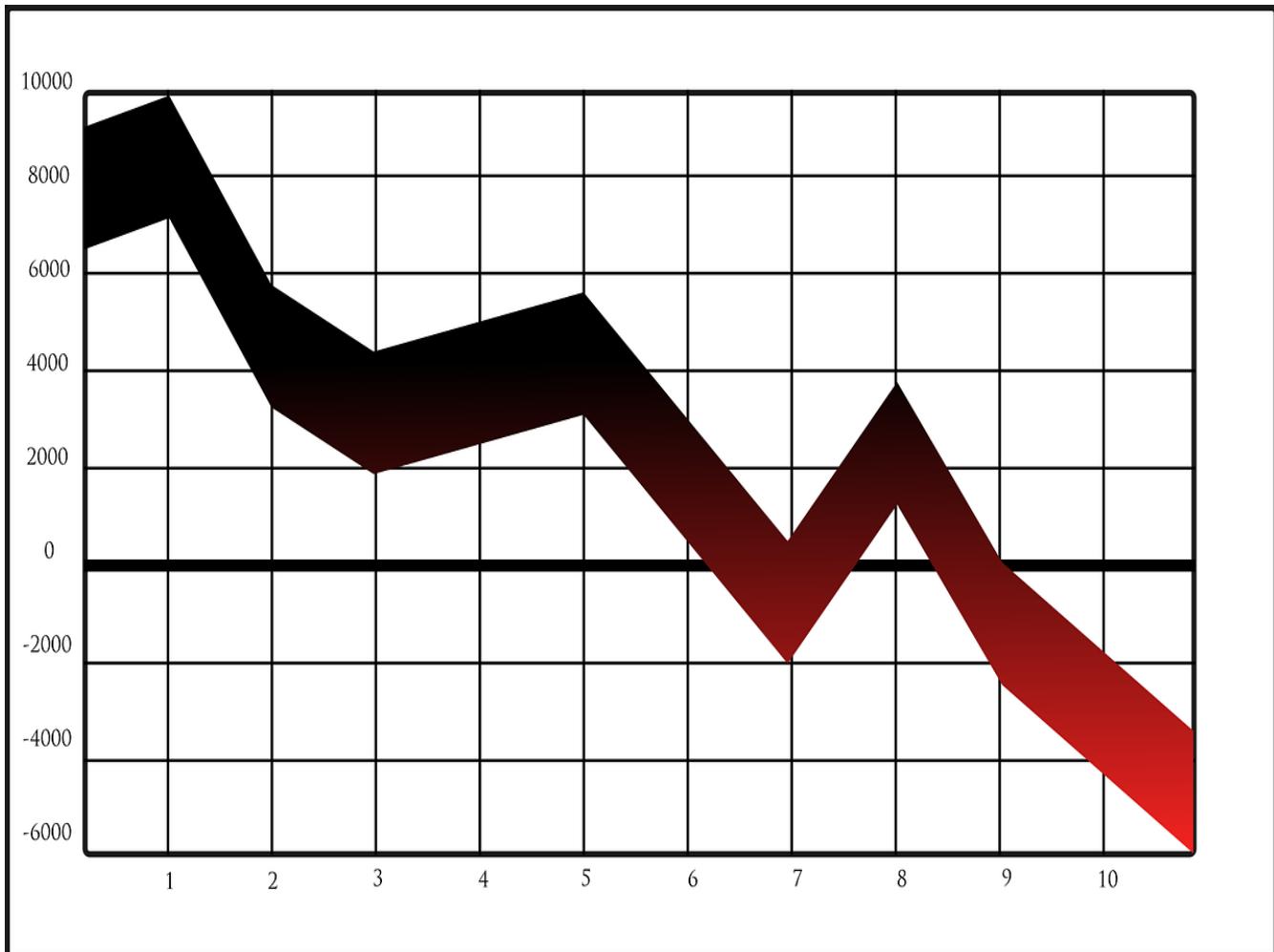
Reference 5



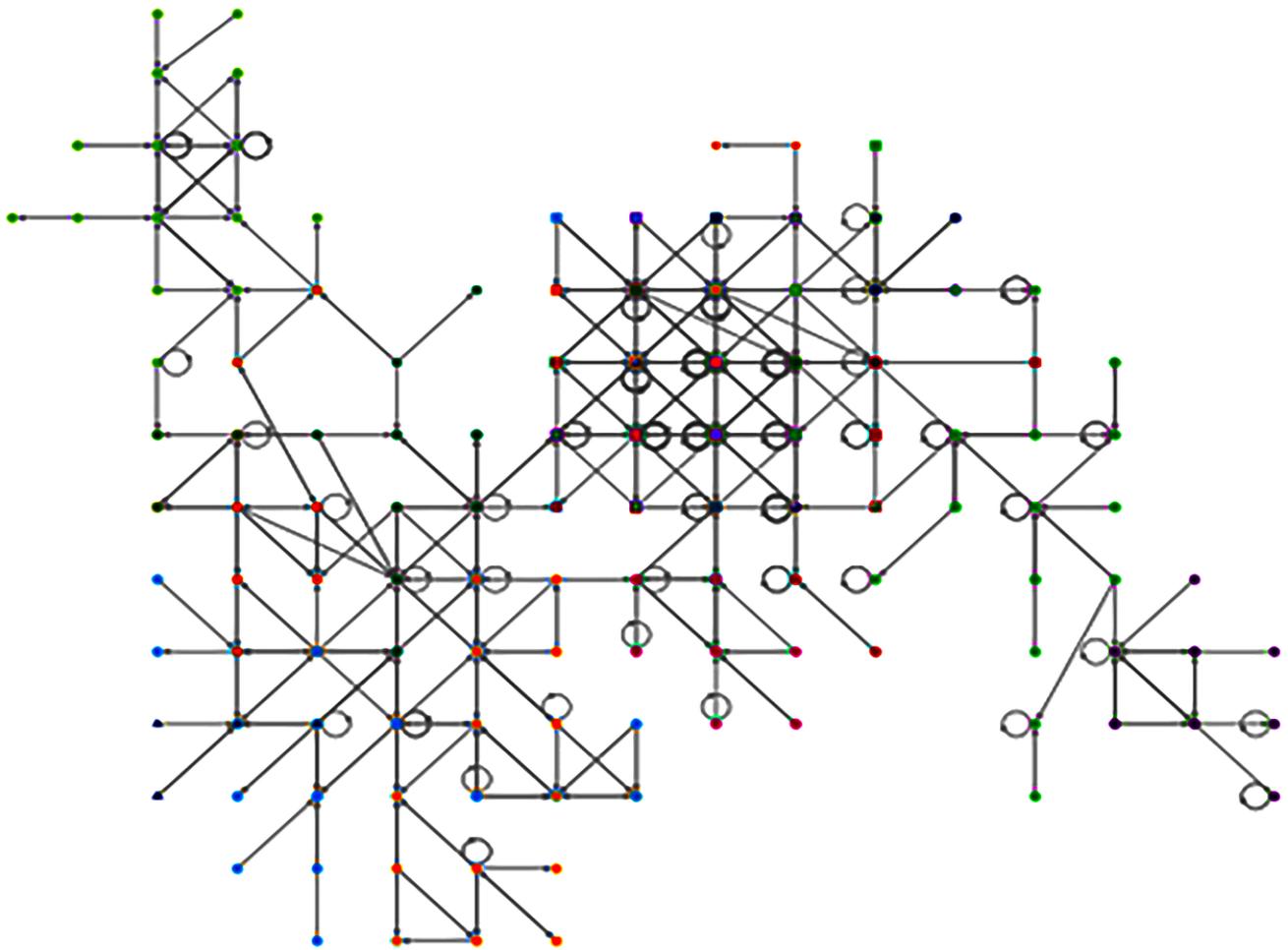
Reference 6



Reference 7



Reference 8



Reference 9

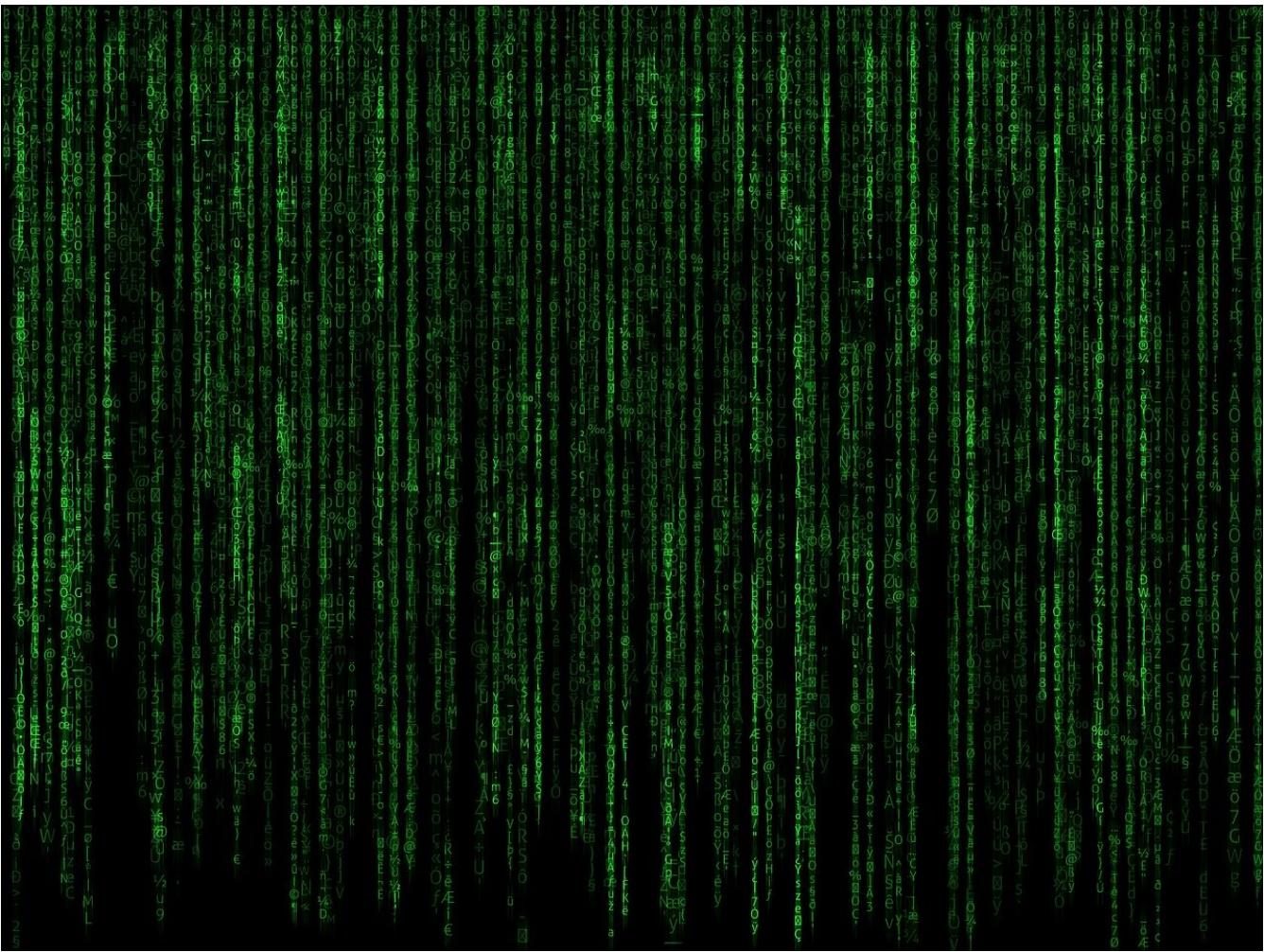


Reference 10

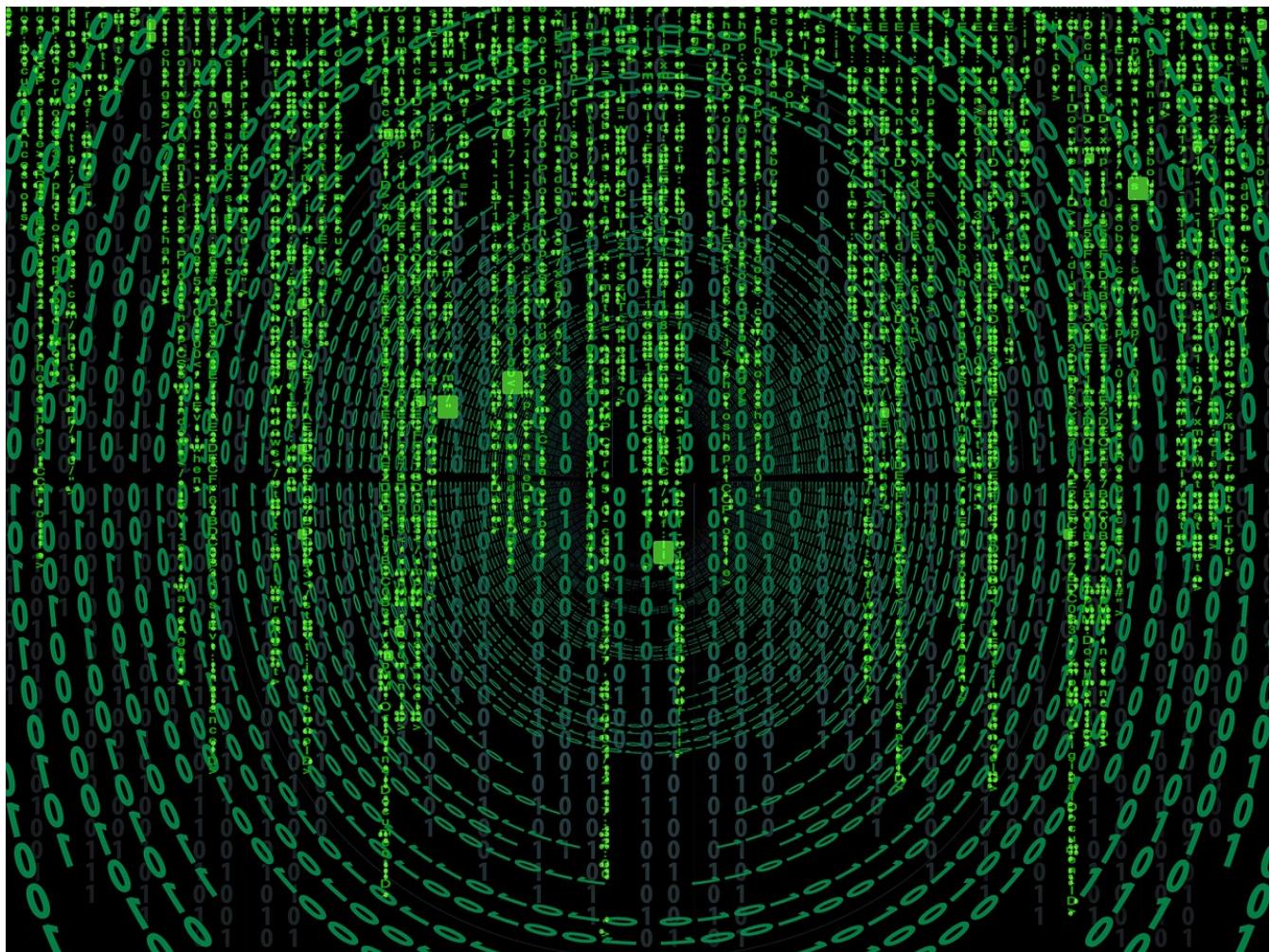
```
1. Sep 15:53 .
2. Sep 15:53 ..
3. Sep 2015 bin -> usr/bin
4. Sep 09:31 boot
5. Sep 15:50 dev
6. Sep 09:32 etc
7. Sep 15:52 home
8. Sep 2015 lib -> usr/lib
9. Sep 2015 lib64 -> usr/lib
10. Jul 10:01 lost+found
11. Aug 22:45 mnt
12. Sep 2015 opt
13. Sep 15:52 private -> /home/encrypted
14. Sep 08:15 proc
15. Aug 15:37 root
16. Sep 15:50 run
17. Sep 2015 sbin -> usr/bin
18. Sep 2015 srv
19. Sep 15:51 sys
20. Sep 15:45 tmp
21. Sep 15:39 usr
22. Jul 10:25 var
root 4096 21. Sep 15:53
root 4096 21. Sep 15:53
root 4096 21. Sep 15:53
```

Your personal files are encrypted

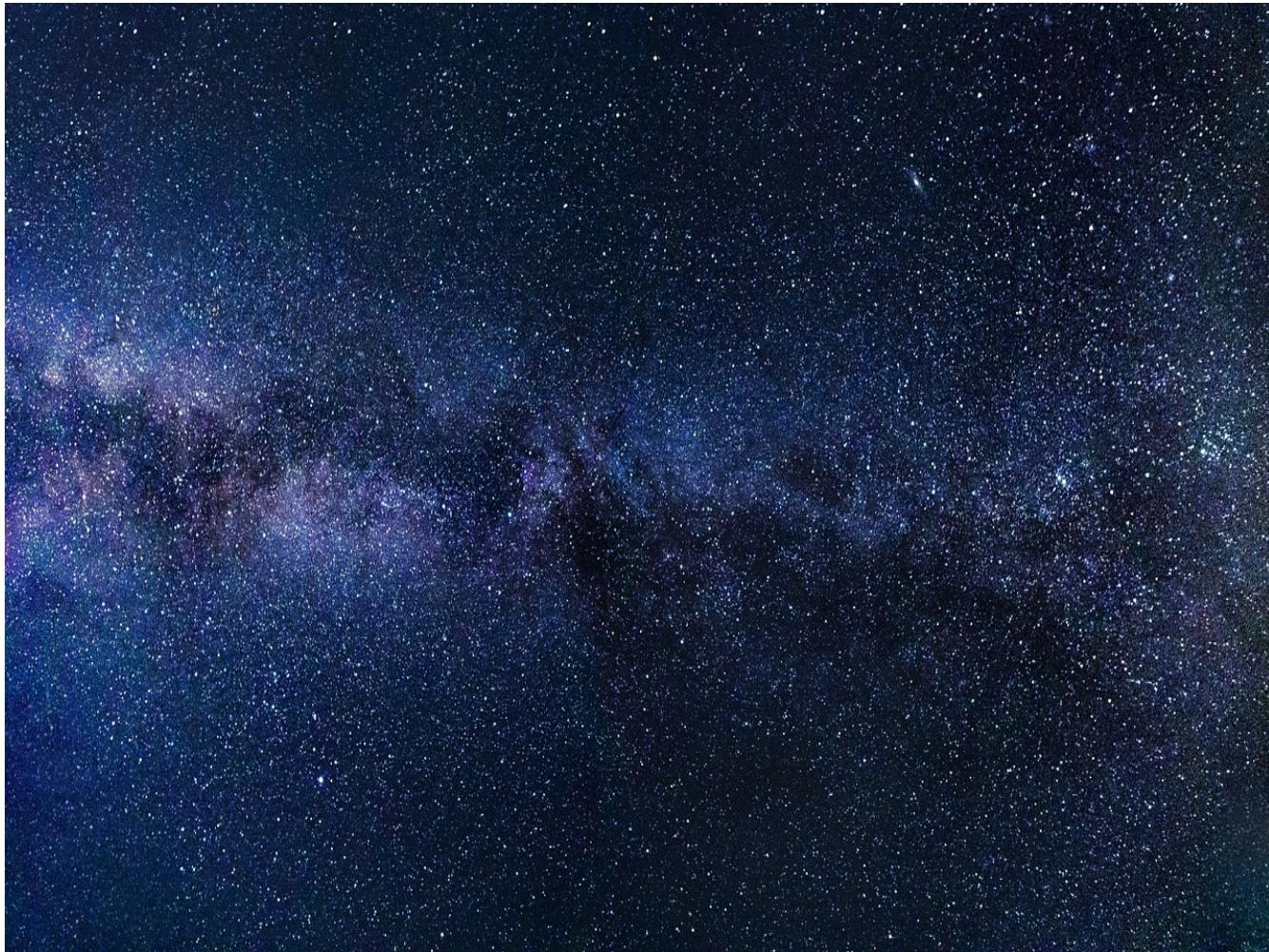




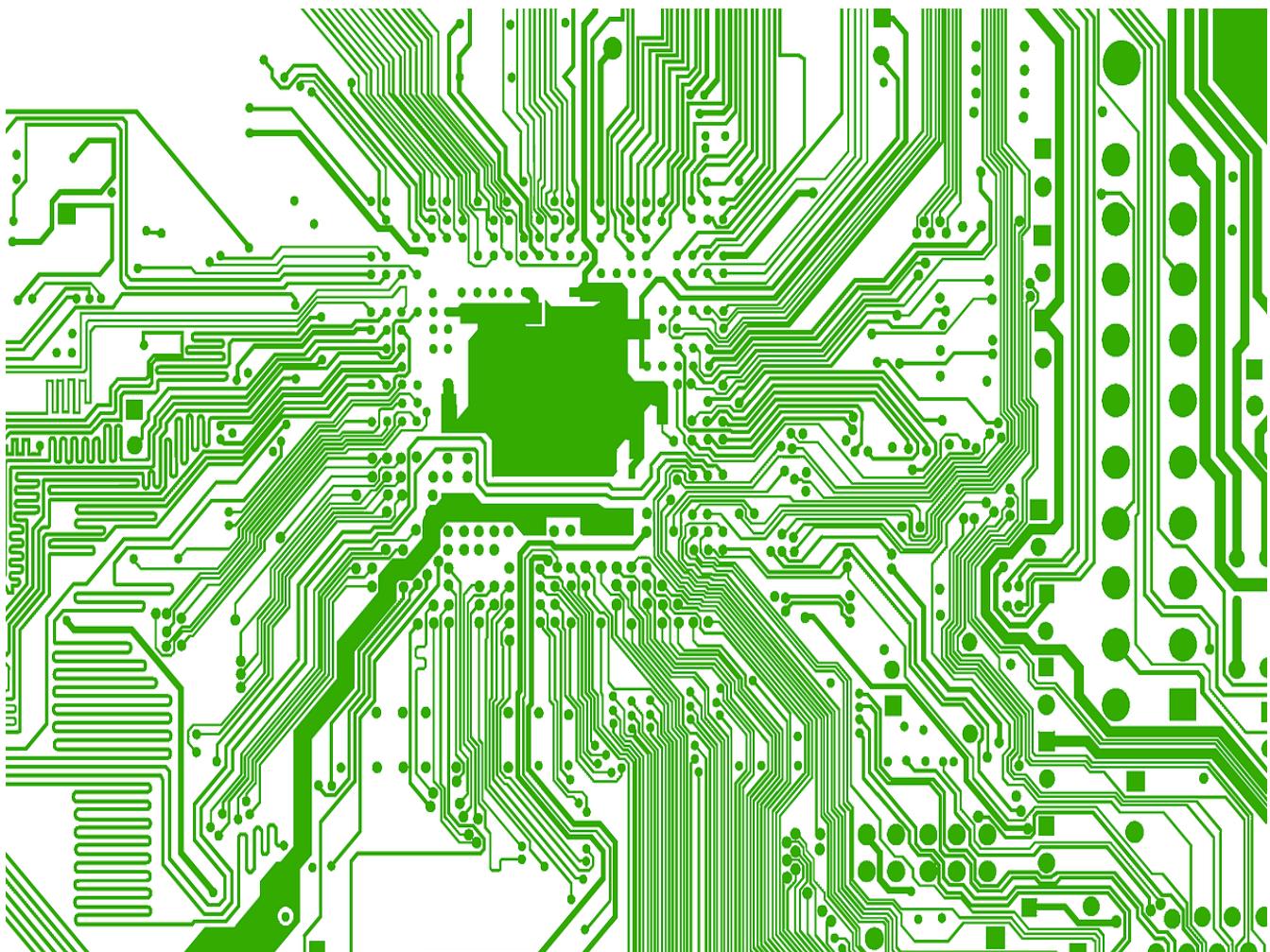
Reference 13



Reference 14



Reference 15



Reference 16

```

744
745     );
746   }
747 }
748
749 $sort_order = array();
750
751 foreach ($quotes as $key => $value) {
752   $sort_order[$key] = $value['sort_order'];
753 }
754
755 array_multisort($sort_order, SORT_ASC, $quotes);
756
757 $this->session->data['lpa']['shipping_methods'] = $quotes;
758 $this->session->data['lpa']['address'] = $address;
759
760 if (empty($quotes)) {
761   $json['error'] = $this->language->get('
762     error_no_shipping_methods');
763 } else {
764   $json['quotes'] = $quotes;
765 }
766
767 if (isset($this->session->data['lpa'][['shipping_method']] && !
768   empty($this->session->data['lpa'][['shipping_method']] &&
769   isset($this->session->data['lpa'][['shipping_method']]['code'])
770 )) {
771   $json['selected'] = $this->session->data['lpa'][[
772     shipping_method]][['code']];
773 } else {
774   $json['selected'] = '';
775 }
776
777 } else {
778   $json['error'] = $this->language->get('error_shipping_methods');
779 }
780
781 $this->response->addHeader('Content-Type: application/json');

382 // || (this.paused = function (e) {
383 //   if (this.$element.find('.next, .prev').length && $support.transition) {
384   this.$element.trigger($.support.transition.end)
385   this.cycle(true)
386 }
387
388 this.interval = clearInterval(this.interval)
389 return this
390 }
391
392 Carousel.prototype.next = function () {
393   if (this.sliding) return
394   return this.slide('next')
395 }
396
397 Carousel.prototype.prev = function () {
398   if (this.sliding) return
399   return this.slide('prev')
400 }
401
402 Carousel.prototype.slide = function (type, next) {
403   var $active = this.$element.find('.item.active')
404   var $next = next || this.getItemForDirection(type, $active)
405   var isCycling = this.interval
406   var direction = type == 'next' ? 'left' : 'right'
407   var fallback = type == 'next' ? 'left' : 'right'
408   var that = this
409
410   if (!$next.length) {
411     if (!this.options.wrap) return
412     $next = this.$element.find('.item')[fallback]()
413   }
414
415   if (!$next.hasClass('active')) return (this.sliding = false)
416
417   var relatedTarget = $next[0]
418   var slideEvent = $.Event('slide.bs.carousel', {
419     relatedTarget: relatedTarget,
420     direction: direction
421   })
422
423 this.$element.trigger(slideEvent)

```

Reference 17





Reference 19

```
if ($window.scrollTop() > header0_initialDistance) {
    header0.css('padding-top', '' + header0_initialPadding + 'px');
}
else {
    header0.css('padding-top', '' + header0_initialPadding - 20);
}

if ($window.scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) > header1_initialPadding) {
        header1.css('padding-top', '' + $window.scrollTop() - header1_initialDistance);
    }
}
else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($window.scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) > header2_initialPadding) {
        header2.css('padding-top', '' + $window.scrollTop() - header2_initialDistance);
    }
}
else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
}
```

Reference 21

Reference 22

Reference 23