

BANNRDEVS HTML5 BANNER AD

Creative Requirements | Ad Edit/Modification

THIS BRIEFING WILL REVIEW GUIDELINES AND REQUIREMENTS FOR HTML5 BANNER AD MODIFICATION.

We recommended you to read HTML5 Banner ads manual to have clear idea about all the aspects of banner customization here you will find the documentation on how to edit, update, and use the banner. Cheers !

Summary

1. Asset Files and Folder structure
2. Coding Guidelines Chnage and Replace
3. Graphic files edit and replace

01

ASSET FILES AND FOLDER STRUCTURE

GUIDELINES

All HTML5-based assets must follow the standard code and structure.

The maximum file size of the .zip file for HTML5 creative is 200 KB, but it should be as light as possible to help with load time.

This does not include the backup image, which is an additional 40 KB.

Supported file types for HTML5 zip:

- HTML/Text: HTML, HTM, CSS, JS
- Images: JPG, JPEG, GIF, PNG, SVG
- Fonts: DFONT, EOT, OTF, TTE, TTF, WOFF, WOFF2

The maximum number of files per .zip for HTML5 creatives is 100. Images, .css, js, index.html should be contained within the same folder. External URLs must reference a secure HTML location (<https://...>) Banners can be produced via hand coding or HTML5 Editor.

HTML5 BANNER FOLDER STRUCTURE

The “.zip” should include images, fonts, .css , js and any other files that are referenced by the HTML file.

Except for the font files, all other files should follow the added descriptor.

The total weight of all files in the HTML5 zip must be 200KB or less. Please note, this does not include the backup image. The backup image has a 40KB file size limit.

DO NOT INCLUDE the following in the HTML5 zip:

- .zip within .zip
- Backup assets; this will be uploaded separately.
- Local or session storage
- Unreferenced files
- % in the name

It is recommended that you use a name that uniquely identifies the banner and will allow you to recognize the banner quickly and easily. Revive Adserver has been built to specifically support HTML5 banners. We recommend that you read BannerDev’s banner manual to get a clear idea of all the aspects of banner customization.

Here we based 300x250 file for documentation basically all code are same

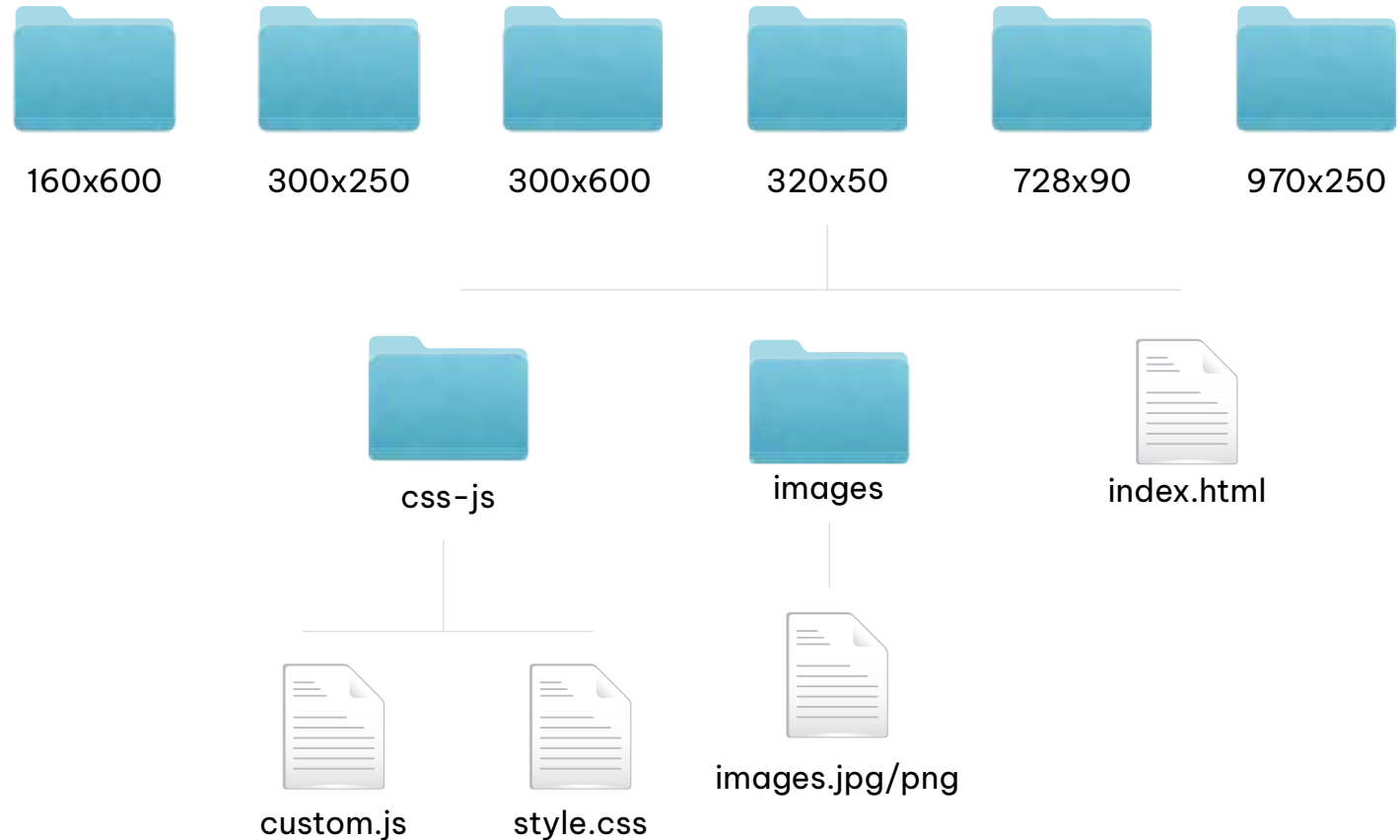
FOLDER STRUCTURE

Extract or unzip by using any tools (xxxxxxxx.zip) file which you downloaded from bannerdevs provided link. You will see following folder structure and need work with those folder. All source file keep in source folder and preview folder for a preview version of all banner



FOLDER STRUCTURE FOR HTML/JS/CSS

Here we based on 300x250 file for demnostrate how to edit and change .html,. css, .js files



02

CODING GUIDELINES CHANGE AND REPLACE

CODING GUIDELINES

All banners must contain Campaign Manager-compliant clickTags; additional information can be found in the Coding Guidelines section.

DO NOT hard code URLs into the HTML clickTag Variable

Use the size meta tag to indicate the intended size for your creative.
Example: `<meta name="ad.size" content="width=300, height=250">`
Make sure to utilize animations and coding that will render in all browsers.

Do not add backup/polite image code into the HTML

EDIT HTML SOURCE

The index.html file is the master file for the banner. To edit the html, you have to open it in any code editor like Bracket, Sublime, Vs Code, or Dreamweaver.

Below, we describe how to edit this file.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   1 <meta name="ad.size" content="width=300,height=250">
7   <title>bannerdevs || 300x250</title>
8   2 <link rel="stylesheet" href="css-js/style.css">
9   <script type="text/javascript">
10    3   var clickTag = "https://www.bannerdevs.com";
11  </script>
12 </head>
```

- 01.** ad size - Important code for a banner ad. Google display network evaluates a banner based on the size of the ad. The ad size must correspond to the banner dimension. The ad size will be 300x250, just like the banner size.
<meta name="ad.size" content="width=300,height=250">
- 02.** All CSS properties are defined in this internal CSS linked file.
- 03.** Click TAG variable

04. ad div banner ads start from here.

05. frame_start properties are defined in this div. You can add or delete banner element and frame.

```
13 <body>
14 <div id="ad">
15 <div id="banner">
16 <!-- ===== Frame One ===== -->
17 <div id="frame_one">
18 
19 <div class="mask">
20 
21 
22 
23 
24 
25 
26 
27 </div>
28 <div class="mask1">
29 
30 </div>
31 
32 </div>
```

06. Frame Start from here

07. Class name for gsap animation

08. Replay button

09. Javascript click tag function for tigger click tag

```
41      <!-- ===== Frame Three ===== -->
42      <div id="frame_three">
43      <div class="mask">
44          
45          
46      </div>
47      
8      
49      </div>
9      <a href="javascript:void(0);" onClick="window.open(window.clickTag)" class="clicktag"></a>
51      </div>
52  </div>
10  <script src="https://s0.2mdn.net/ads/studio/cached_libs/
54      tweenmax_1.18.0_499ba64a23378545748ff12d372e59e9_min.js"></script>
11  <script src="css-js/custom.js"></script>
56  </body>
57  </html>
```

10. GSAP cdn library link (GSAP is a JavaScript library for building high-performance animations that work in <https://greensock.com/>)

11. Custom javascript for animation

CSS SOURCE CODE

Internal .css file located css-js folder all style using HTML5 banner put here

```
1  /*-----  
2      Reset  
3  -----*/  
4  * {  
5      margin: 0;  
6      padding: 0;  
7      position: absolute;  
8  }  
9  
10 #banner * {  
11     position: absolute;  
12 }  
13  
14 #banner br {  
15     position: static;  
16 }  
17  
18 img {  
19     border: none;  
20 }  
21
```

01. Universal property for all element

02. Image property

03. Ad size define

04. Banner all property set here

```
22  /*-----  
23      Layout  
24  -----*/  
25  
3  #ad {  
27     width: 300px;  
28     height: 250px;  
29     -moz-user-select: -moz-none;  
30     -khtml-user-select: none;  
31     -webkit-user-select: none;  
32     -ms-user-select: none;  
33     user-select: none;  
34 }  
35  
4  #banner {  
37     position: absolute;  
38     display: block;  
39     background: #ffffff;  
40     width: 300px;  
41     height: 250px;  
42     overflow: hidden;  
43     cursor: pointer;  
44     visibility: hidden;  
45     border: 1px solid #666;  
46     -webkit-box-sizing: border-box;  
47     box-sizing: border-box;  
48 }  
49
```

05. click tag variable set

06. All Frame property

```
5 50 .clicktag {  
51     width: 100%;  
52     height: 100%;  
53     z-index: 10;  
54     cursor: pointer;  
55 }  
56  
57 57 .clicktag img {  
58     border: none;  
59     outline: none;  
60 }  
61  
6 62 #frame_one, #frame_two, #frame_three{  
63     width: 100%;  
64     height: 100%;  
65 }  
66
```

07. Replay button property

```
7 73 #replay{  
96     z-index: 999;  
97     width: 14px;  
98     height: 14px;  
99     opacity: 0;  
100    left: 6px;  
101    top: 5px;  
102    cursor: pointer;  
103 }
```

JAVASCRIPT SOURCE CODE

Internal .css file located css-js folder all style using ib banner put here

01. Initial banner property id

02. Master time line initial for banner start

```
1 var banner = document.getElementById('banner');  
2 var ctaAnimation = false;  
3  
4  
5 var masterTimeline = new TimelineMax({paused:true, repeat:1, repeatDelay:2.5}),  
6 t1 = new TimelineMax();  
7 //Timeline animation...  
8 t1.set(banner, {visibility: "visible"})  
9 .add(function(){if(ctaAnimation){TweenMax.set(".cta, .cta_hover", {alpha:0})}})  
10 .add(function(){ctaAnimation = false;})  
11 .set(".cta_hover", {alpha:0, display:"none", visibility:"hidden"})
```

03. Repeat property how many time a banner repeat 0 for no repeat

04. Repeat dealy how many times a banner stay before start again

05. t1 to set new time line

06. Banner visibility set visible

07. CTA hover function

08. Set default CTA button display none

09. Add frame here

10. Class property for animation

11. ease property for animation

```
13  /*frame one */
9   .add("frame1")
15  .from(".gift1", .3, {y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame1")
16  .from(".gift2", .3, {y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame1-.1")
17  .from(".gift3", .3, {y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame1")
18  .from(".christmass", .5, {alpha:0, y:10, ease:Elastic.easeOut}, "frame1+-.2")
19  .from(".sale", .3, {alpha:0, y:-10, ease:Elastic.easeOut}, "frame1+-.3")
20  .from(".upto", .3, {alpha:0, y:-10, ease:Elastic.easeOut}, "frame1+-.4")
21  .from(".price", .3, {alpha:0, y:-10, ease:Elastic.easeOut}, "frame1+-.5")
22  .from(".enjoy", .3, {alpha:0, y:-10, ease:Elastic.easeOut}, "frame1+-.5")
23  .to(".gift2,.gift1,.gift3,.grass", .3, {y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame1+=3")
24  .from(".cap", .3, {y:-70, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame1+=3")
25
```

12. Frame name to ad time

13. How many time stay per frame

```
32  /*frame three */
33  .add("frame3", "frame2+=3")
34  .from(".product2", .3, {y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame3")
35  .from(".violator2", .3, {scale:0, transformOrigin:"bottom right", rotation:0.02, ease:Sine.easeOut}, "frame3+-.2")
36  .to(".enjoy", .3, {y:50, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame3")
37  .from(".cta", .3, {alpha:0, y:250, rotation:0.02, ease:Circ.easeOuteaseOut}, "frame3+=2.5")
38  .to("#replay", 0.2, {alpha:1, rotation: -360, onComplete:function(){ctaAnimation = true;}}, "frame3+=2.5")
39  ;
40
```

14. Frame 2nd frame start

15. Cta animation function

15. Master timeline add

16. Cta animation function for mouseover

17. Cta animation function for mouseout

```
41 /*-----*/
42 Add all timelines to master timeline...
43 -----*/
15 masterTimeline.add([t1]);
45
46
16 banner.addEventListener("mouseover", function(){
  if(ctaAnimation){
49 TweenMax.to(".cta", .5, {scale: 1.2,
50   transformOrigin: "50% 50%", ease:Elastic.easeOut});
51
52 }
53 })
54
17 banner.addEventListener("mouseout", function(){
55   if(ctaAnimation){
56
57     TweenMax.to(".cta", .5, {scale: 1,
58       transformOrigin: "50% 50%", ease:Elastic.easeOut});
59
60   }
```



```
63  /*-----  
64   After window load...  
65  -----*/  
18  window.onload = function()  
67  {  
68     masterTimeline.play();  
69  }  
70  
71  
72  /*-----  
73   Count animation duration of a timeline...  
74  -----*/  
19  var currentDuration = masterTimeline.totalDuration();  
76  console.log(currentDuration);  
77  
20  replay.addEventListener("click", function(){  
79     masterTimeline.restart();  
80  })  
81
```

18. Window on load function

19. Total duration calculation

20. Replay function for banner animation

03

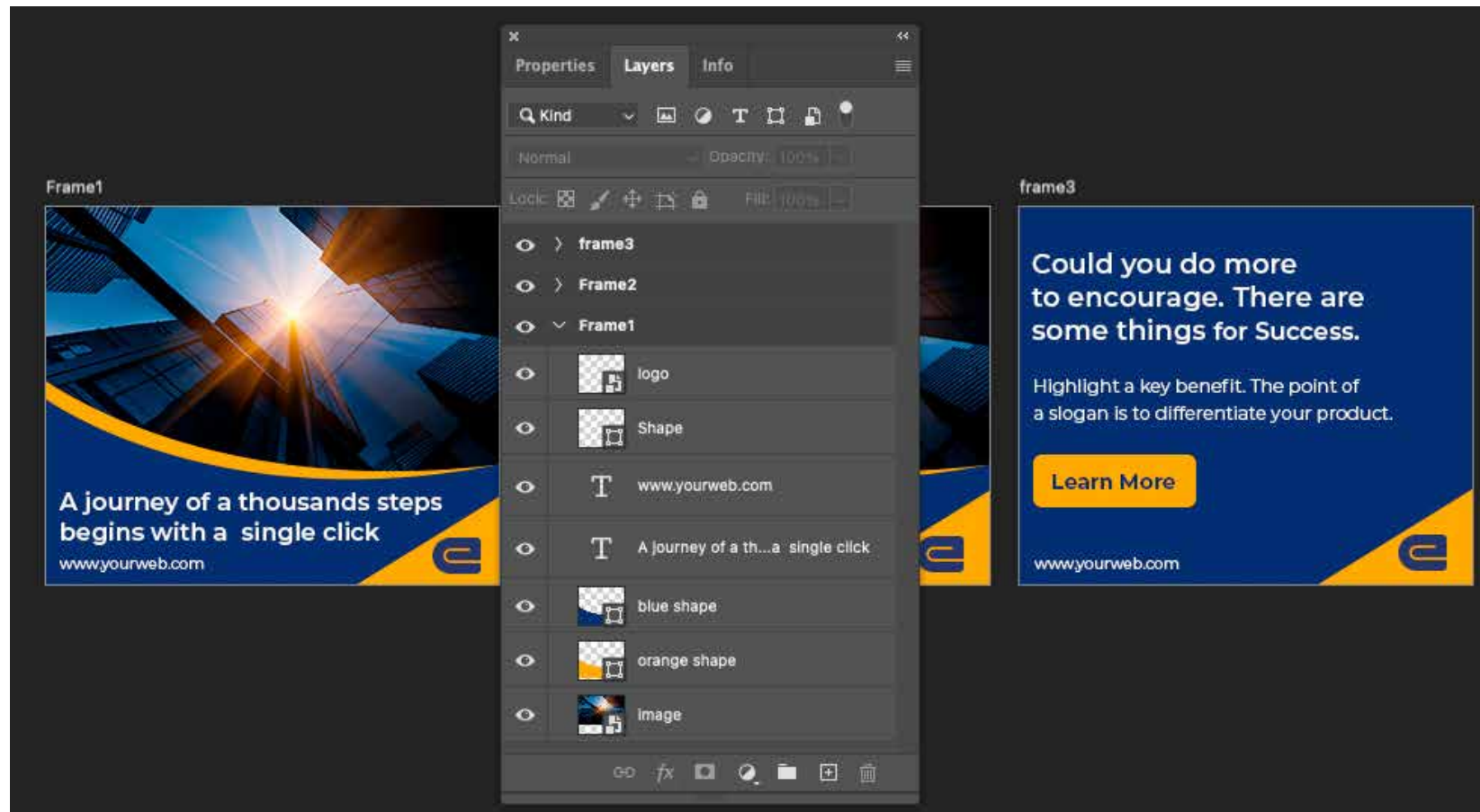
**GRAPHIC FILES
EDIT AND
REPLACE**

GUIDELINES FOR PSD/GRAPHIC FILE EDIT

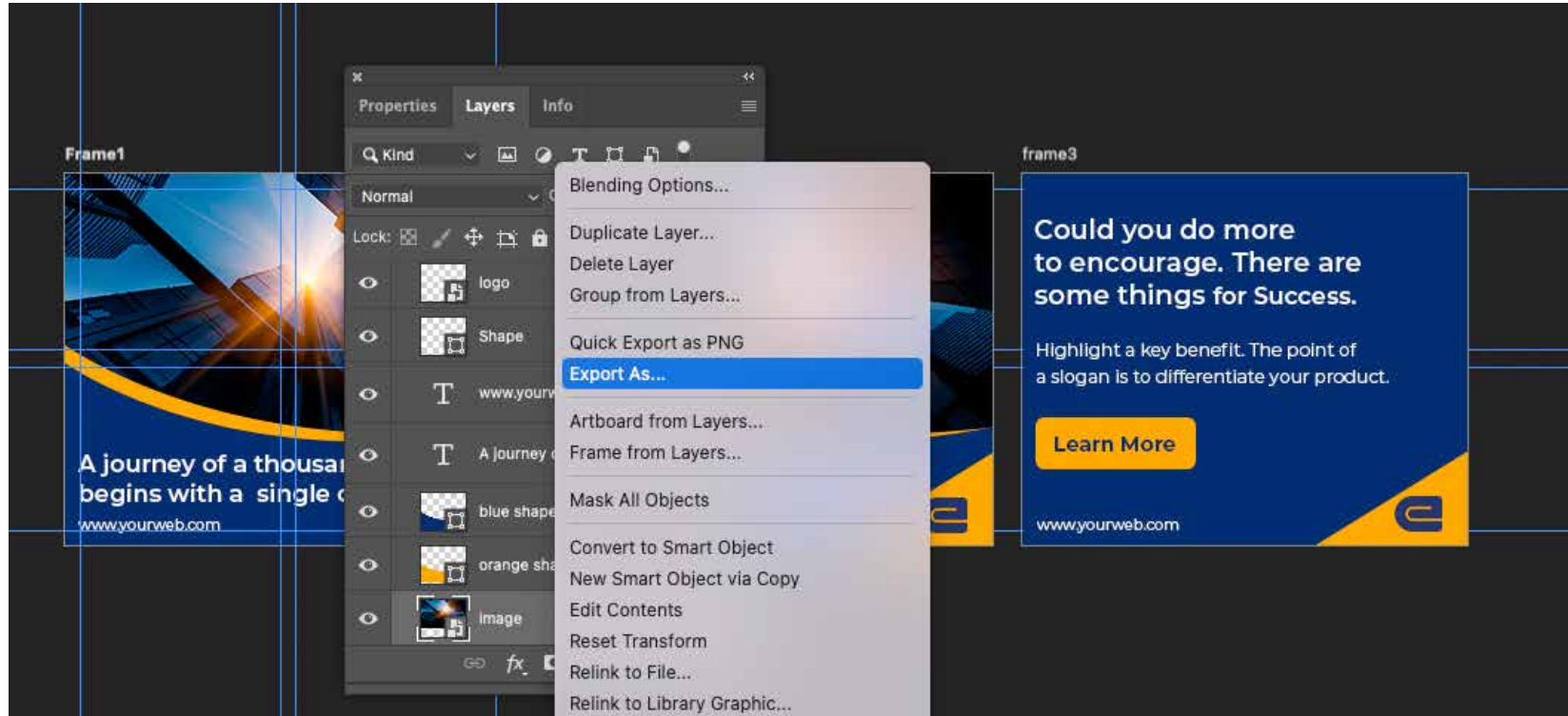
Required All files you can find specific folder Because PSD, ai, and other graphic source files are editable, files can be easily edited and replaced. Fonts are available on Google. <https://fonts.google.com/>

EDIT PSD SOURCE

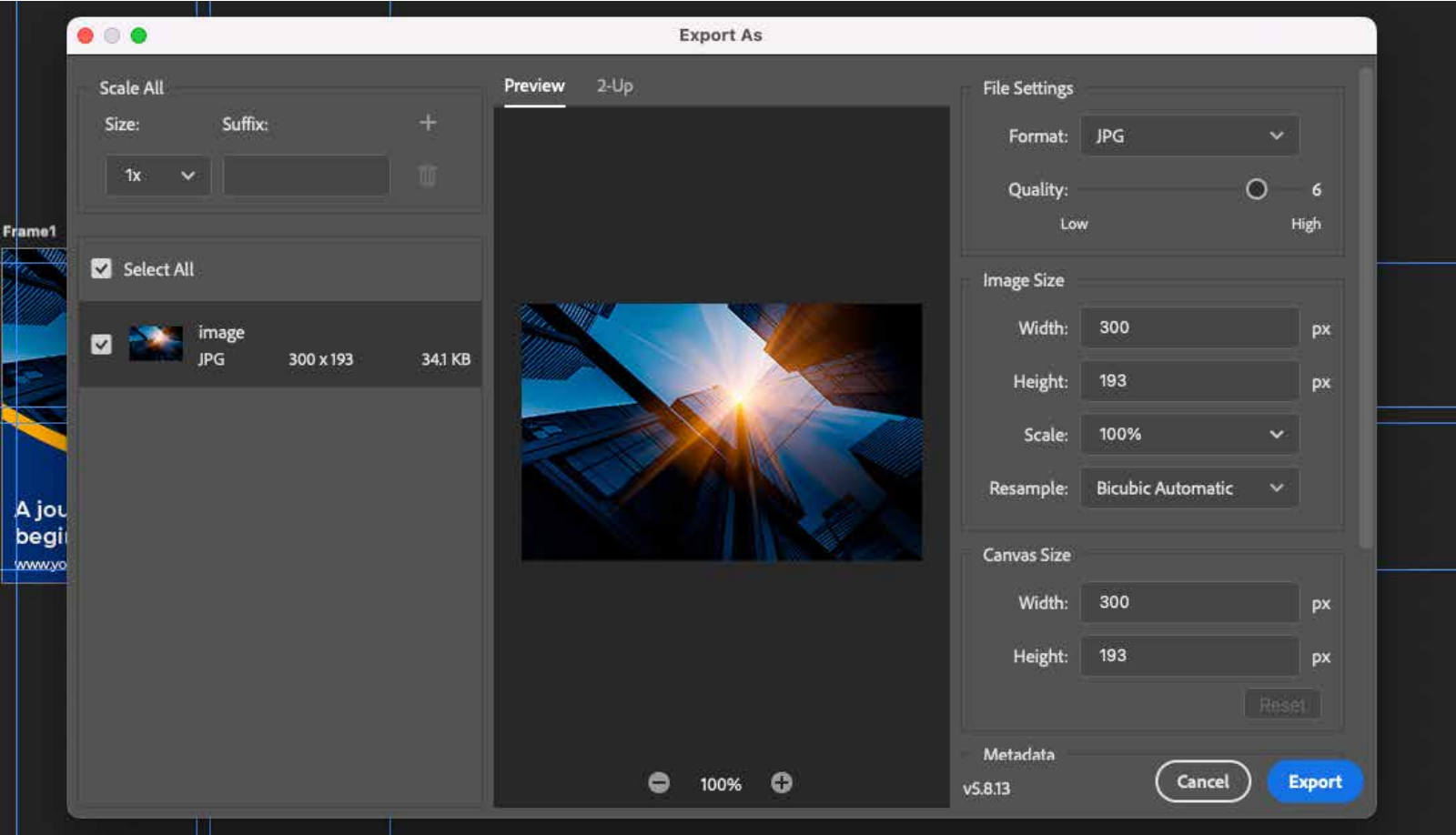
All frame are layer based All text and images are easy to edit you can change it very easily



Export image from photoshop file



Save the image as specific images folder



THANK YOU

HOPE THIS HELP YOU.
IF ANY PROBLEM FEEL FREE
TO CONTACT WITHUS AT :
infobannerdevs@gmail.com

