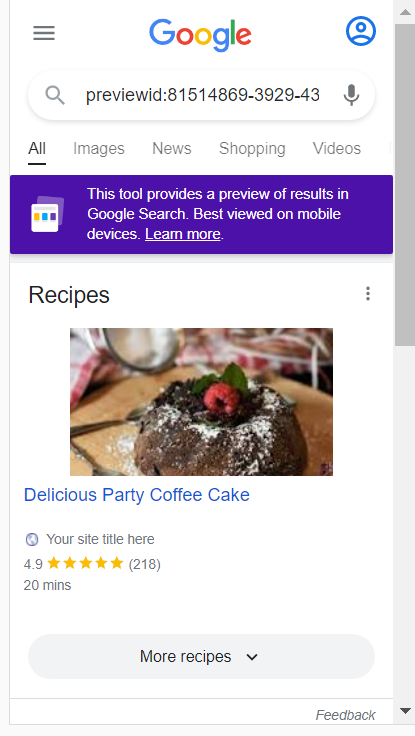
**Schema Object using MicroData format,RDFa,JSON-ld**

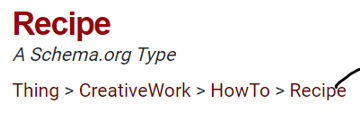
NOTE: here we are going to just cover coding part, no theory here, if u want to see the theory please see the another word file where I explained the theory part

1. **MicroData**
2. <!**DOCTYPE** html>
3. <**html** lang="en">
4. <**head**>
5. <**meta** charset="UTF-8" />
6. <**meta** name="viewport" content="width=device-width, initial-scale=1.0" />
7. <**meta** http-equiv="X-UA-Compatible" content="ie=edge" />
8. <**link**
9. rel="stylesheet"
10. href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
11. />
12. <**title**>Delicious Party Coffee Cake</**title**>
13. </**head**>
14. <**body** class="mt-3">
15. <!-- Page Content -->
16. <**main** class="container" itemscope itemtype="https://schema.org/Recipe">
17. <!-- Article Section -->
18. <**article**>
19. <!-- // explaniation
20. itemscuope is scope of the item , till the tag ends
21. -->
22. <!--
23. itemtype is the type of caterofry we are going to use in our scheme, currenntly we are using the Recipe so we keep the recipe
24. -->
25. <**header**>
26. <**h1** itemprop="name" class="text-success">
27. <!-- here itemprop means the property of the item(Recipe currently) -->
28. Delicious Party Coffee Cake
29. </**h1**>
30. <**p**>
31. Recipe author:
32. <**span**
33. itemscope
34. itemtype="https://schema.org/Person"
35. itemprop="author"
36. >
37. <**span** itemprop="name"> Mary Stone </**span**>
38. </**span**>
39. <!-- itemprop of recipe author -->
40. </**p**>
41. <**p**>
42. <**meta** itemprop="datePublished" content="2021-09-28" />
43. Date published: March 16, 2020
44. <!-- dataof publisted which present in the recipe property -->
45. </**p**>
46. <**p** itemprop="description">
47. This coffee cake is awesome and perfect for parties.
48. </**p**>
49. <**img**
50. itemprop="image"
51. src="https://cdn.pixabay.com/photo/2019/09/24/21/49/cake-4502270\_1280.jpg"
52. height="300"
53. alt="Party Coffee Cake"
54. />
55. </**header**>
56. <**section** class="mt-4">
57. <!-- recipeYield -->
58. <**h5**>Yield: <**span** itemprop="recipeYield"> 10 servings </**span**></**h5**>
59. <!-- https://en.wikipedia.org/wiki/ISO\_8601 -->
60. <!-- P3Y6M4DT12H30M5S -->
61. <**h5**>
62. Prep time:
63. <**meta** itemprop="prepTime" content="PT20M" />
64. 20 minutes
65. </**h5**>
66. <**h5**>
67. Cook time:
68. <**meta** itemprop="cookTime" content="PT20M" />
69. 30 minutes
70. </**h5**>
71. <**h5**>
72. Total time:
73. <**meta** itemprop="totalTime" content="PT20M" />
74. 150 minutes
75. </**h5**>
76. <**h5**>
77. Category:
78. <**span** itemprop="recipeCategory"> Cake </**span**>
79. </**h5**>
80. <**h5**>
81. Recipe Cuisine:
82. <**span** itemprop="recipeCuisine"> French </**span**>
83. </**h5**>
84. <**h2** class="text-success">Ingredients</**h2**>
85. <**ul**>
86. <**li** itemprop="recipeIngredient">2 cups of flour</**li**>
87. <**li** itemprop="recipeIngredient">3/4 cup white sugar</**li**>
88. <**li** itemprop="recipeIngredient">2 teaspoons baking powder</**li**>
89. <**li** itemprop="recipeIngredient">1/2 teaspoon salt</**li**>
90. <**li** itemprop="recipeIngredient">1/2 cup butter</**li**>
91. <**li** itemprop="recipeIngredient">2 eggs</**li**>
92. <**li** itemprop="recipeIngredient">3/4 cup milk</**li**>
93. <**li** itemprop="recipeIngredient">1 teaspoon vanillaextract</**li**>
94. <**li** itemprop="recipeIngredient">1 1/2 teaspoon group cinnamon</**li**>
95. </**ul**>
96. <**h2** class="text-success">Directions</**h2**>
97. <**ul**>
98. <**li**
99. itemprop="recipeInstructions"
100. itemscope
101. itemtype="https://schema.org/HowToStep"
102. >
103. <**span** itemprop="text">
104. Preheat the oven to 350 degrees F. Grease and flour a 9x9 inch
105. pan.
106. </**span**>
107. </**li**>
108. <**li**
109. itemprop="recipeInstructions"
110. itemscope
111. itemtype="https://schema.org/HowToStep"
112. >
113. <**span** itemprop="text">
114. Make the streusal topping. In a medum bowl, combine flour,
115. sugar, and cinnamon. Mix in butteruntil the entire mixture is
116. crumbly.
117. </**span**>
118. </**li**>
119. <**li**
120. itemprop="recipeInstructions"
121. itemscope
122. itemtype="https://schema.org/HowToStep"
123. >
124. <**span** itemprop="text">
125. In a large bowl, combine flour, sugar, baking powder, and salt.
126. Mix in the butter. Mix in the milk, egg, and vanilla. Spread
127. into the prepared pan.
128. </**span**>
129. </**li**>
130. <**li**
131. itemprop="recipeInstructions"
132. itemscope
133. itemtype="https://schema.org/HowToStep"
134. >
135. <**span** itemprop="text">
136. Sprinkle the streudal mixture on top of the cake.
137. </**span**>
138. </**li**>
139. <**li**
140. itemprop="recipeInstructions"
141. itemscope
142. itemtype="https://schema.org/HowToStep"
143. >
144. <**span** itemprop="text">
145. Bake for 30 to 35 minutes, or until firm. Allow to cool.
146. </**span**>
147. </**li**>
148. </**ul**>
149. <**h2** class="text-success">Nutrition facts</**h2**>
150. <**p**
151. itemprop="nutrition"
152. itemscope
153. itemtype="https://schema.org/NutritionInformation"
154. >
155. Per slice :
156. <**span** itemprop="calories">2701</**span**> calories;
157. <**span** itemprop="fatContent"> 12</**span**> g fat;
158. <**span** itemprop="carbohydrateContent"> 37</**span**> g carbohydrates;
159. <**span** itemprop="proteinContent">2</**span**> g protein;
160. <**span** itemprop="cholesterolContent"> 45</**span**> mg cholesterol;
161. <**span** itemprop="sodiumContent">1 95</**span**> mg sodium.
162. </**p**>
163. <**h2** class="text-success">Keywords</**h2**>
164. <**p** itemprop="keywords">Coffe Cake, Party Cake</**p**>
165. </**section**>
166. </**article**>
167. <!-- /.row -->
168. <**hr** />
169. <!-- Reviews Section -->
170. <**section**>
171. <**h2** class="text-success">Sample Reviews ★★★★★</**h2**>
172. <**h6**
173. itemprop="aggregateRating"
174. itemscope
175. itemtype="https://schema.org/AggregateRating"
176. >
177. <**span** itemprop="ratingValue">4.9</**span**>
178. out of
179. <**span** itemprop="bestRating"> 5 </**span**>
180. (Total
181. <**span** itemprop="reviewCount">218</**span**>
182. )
183. </**h6**>
184. <!-- list of reviews -->
185. <**article**
186. itemprop="review"
187. itemscope
188. itemtype="https://schema.org/Review"
189. >
190. <**h4** itemprop="author">Mary</**h4**>
191. <**time** datetime="2020-06-06T13:31:04-08:00">
192. <**meta**
193. itemprop="datePublished"
194. content="2020-06-06T13:31:04-08:00"
195. />
196. June 6<**sup**>th</**sup**>, 2020 at 1:31</**time**
197. >
198. <**p**
199. itemprop="reviewRating"
200. itemscope
201. itemtype="https://schema.org/Rating"
202. >
203. Rating: <**span** itemprop="ratingValue">5</**span**>
204. </**p**>
205. <**p**>
206. Review for:
207. <**span** itemprop="itemReviewed">Delicious Party Coffee Cake</**span**>
208. </**p**>
209. <**p** itemprop="reviewBody">This coffee cake is delicious!</**p**>
210. </**article**>
211. <**article**
212. itemprop="review"
213. itemscope
214. itemtype="https://schema.org/Review"
215. >
216. <**h4** itemprop="author">Mary</**h4**>
217. <**time** datetime="2020-06-06T13:31:04-08:00">
218. <**meta**
219. itemprop="datePublished"
220. content="2020-06-06T13:31:04-08:00"
221. />
222. June 6<**sup**>th</**sup**>, 2020 at 1:31</**time**
223. >
224. <**p**
225. itemprop="reviewRating"
226. itemscope
227. itemtype="https://schema.org/Rating"
228. >
229. Rating: <**span** itemprop="ratingValue">5</**span**>
230. </**p**>
231. <**p**>
232. Review for:
233. <**span** itemprop="itemReviewed">Delicious Party Coffee Cake</**span**>
234. </**p**>
235. <**p** itemprop="reviewBody">This coffee cake is delicious!</**p**>
236. </**article**>
237. <**article**
238. itemprop="review"
239. itemscope
240. itemtype="https://schema.org/Review"
241. >
242. <**h4** itemprop="author">Susan</**h4**>
243. <**time** datetime="2020-02-09T13:31:04-08:00">
244. <**meta**
245. itemprop="datePublished"
246. content="2020-06-06T13:31:04-08:00"
247. />
248. Feb 9<**sup**>th</**sup**>, 2020 at 1:31</**time**
249. >
250. <**p**
251. itemprop="reviewRating"
252. itemscope
253. itemtype="https://schema.org/Rating"
254. >
255. Rating: <**span** itemprop="ratingValue">5</**span**>
256. </**p**>
257. <**p**>
258. Review for:
259. <**span** itemprop="itemReviewed">Delicious Party Coffee Cake</**span**>
260. </**p**>
261. <**p** itemprop="reviewBody">
262. The cake instructions are superbly crafted and accurately designed.
263. </**p**>
264. </**article**>
265. <**article**
266. itemprop="review"
267. itemscope
268. itemtype="https://schema.org/Review"
269. >
270. <**h4** itemprop="author">Mark</**h4**>
271. <**time** datetime="2020-07-16T13:31:04-08:00">
272. <**meta**
273. itemprop="datePublished"
274. content="2020-06-06T13:31:04-08:00"
275. />
276. July 16<**sup**>th</**sup**>, 2020 at 1:31</**time**
277. >
278. <**p**
279. itemprop="reviewRating"
280. itemscope
281. itemtype="https://schema.org/Rating"
282. >
283. Rating: <**span** itemprop="ratingValue">5</**span**>
284. </**p**>
285. <**p**>
286. Review for:
287. <**span** itemprop="itemReviewed">Delicious Party Coffee Cake</**span**>
288. </**p**>
289. <**p** itemprop="reviewBody">
290. The cake also includes paprika, which makes me nervous.
291. </**p**>
292. </**article**>
293. <**p**><**a** href="#">(Expand all reviews)</**a**></**p**>
294. </**section**>
295. <!-- /list of reviews -->
296. <!-- Footer -->
297. <**footer** class="mt-5">
298. <**hr** />
299. <**p**>
300. Copyright &copy; Home of the best cakes, 2020 |
301. <**a** href="#" target="\_blank" rel="nofollow">Rahul Giri</**a**>
302. </**p**>
303. </**footer**>
304. </**main**>
305. <!-- /.container -->
306. </**body**>
307. </**html**>

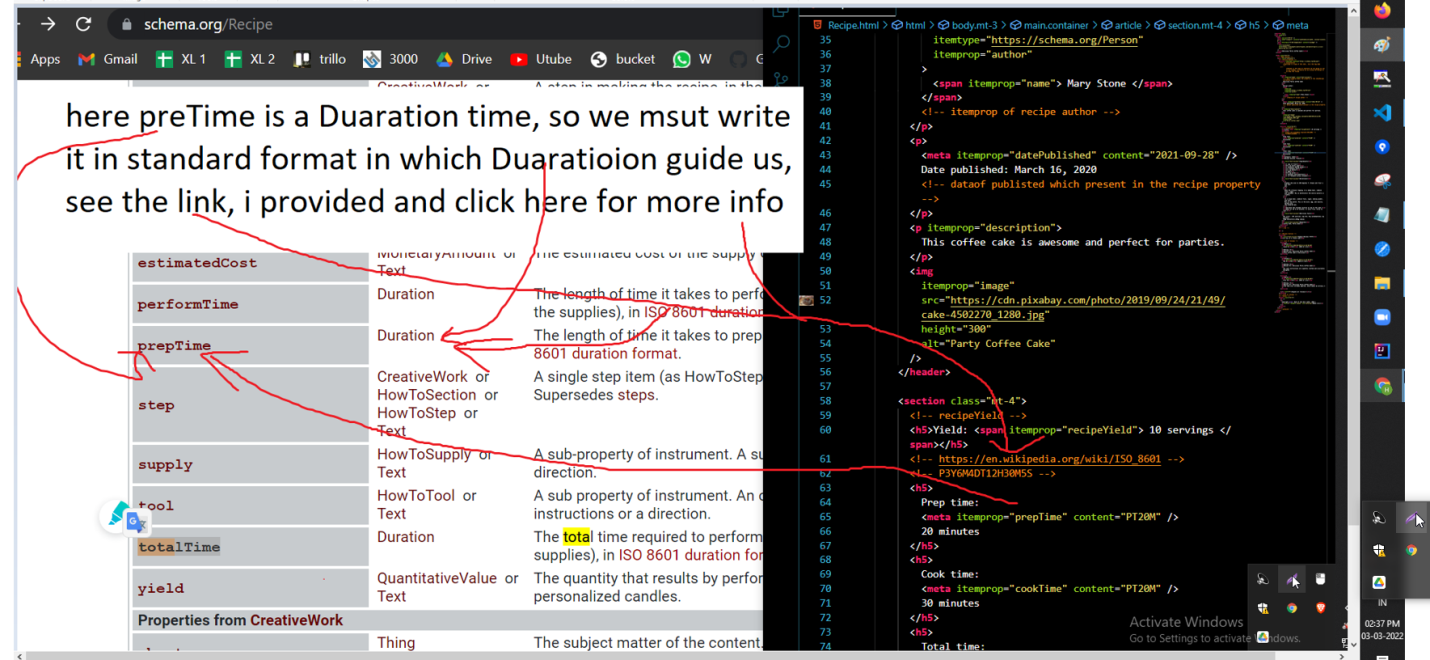
**Output**

****

**Code Explanation**

1. In above example code , itemscope is scope of the item , till the tag ends
2. itemtype is the type of caterogry which we are going to use in our scheme, currently , we are using the Recipe so we keep the recipe
3. here itemprop means the property of the item(Recipe currently)
4. In that we have all the property like description author, name and n-1 property these all property descripe the importance of the recipe to the Google in the form of recipe object
5. When u which the visit the scheme.org official website then u will find something like these  
   

Here Recipe is inheriting all the property from the HowTo and creativeWork and Things i.e now we can use all the property of the HowTo and creativeWork and Things in Recipe , along with recipe property in our code

1. There is no specific rule that we must use these one only or not, according to the our website, we will prepare the scheme object, but here name and images are mandatory because its recommend by goggle
2. NOTE : what we are writing in scheme Object all the properties are not showen in the UI part, just those property are taken by the Google which Google wants but other property also we must and should need to mention in our code so that google has a good understanding of our recipe or any scheme Object
3.   
   please see these photo, its explaning how to write the date format in the scheme Object
4. how do i know that we must use these property only ani, anthi, according the content which we have in our website, we msut search that name in scheme.org website in recipe categroy then we must use that one here anthi

s  
    <**p**

            itemprop="nutrition"

            itemscope

            itemtype="https://schema.org/NutritionInformation"

          >

            Per slice :

            <**span** itemprop="calories">2701</**span**> calories;

            <**span** itemprop="fatContent"> 12</**span**> g fat;

            <**span** itemprop="carbohydrateContent"> 37</**span**> g carbohydrates;

            <**span** itemprop="proteinContent">2</**span**> g protein;

            <**span** itemprop="cholesterolContent"> 45</**span**> mg cholesterol;

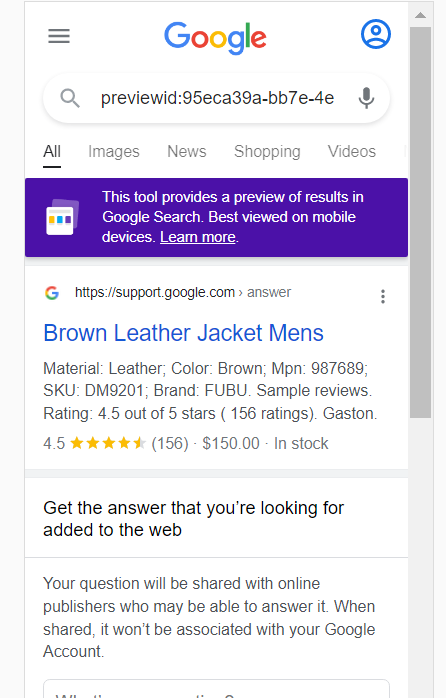
            <**span** itemprop="sodiumContent">1 95</**span**> mg sodium.

          </**p**>

we need to write again scope here becoz these is a another type(nutritionInfomation ), so we need to write

1. For example in our Recipe example, the main mandatory property are name and Image as per the Google guidelines, u may think know do I know that these 2 fields(name and images) are mandatory , plz check the google docs(<https://developers.google.com/search/docs/advanced/structured-data/recipe> ) they mentioned all the mandatory properties and recommened properties in their google docs  
   same rule is applicable for the all the hotel and products and organizations
2. **RDFa format**
3. <!**doctype** html>
4. <**html** lang="en">
5. <**head**>
6. <**meta** charset="utf-8">
7. <**title**>Brown Leather Jacket Mens</**title**>
8. <**meta** http-equiv="X-UA-Compatible" content="ie=edge">
9. <**meta** name="viewport" content="width=device-width, initial-scale=1.0">
10. <**link** rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
11. </**head**>
12. <**body** class="container mt-4">
13. <**main** role="main">
14. <**article** vocab="https://schema.org/" typeof="Product">
15. <**header**>
16. <**h2** class="text-info" property="name">Brown Leather Jacket Men</**h2**>
17. <**img** src="https://images-na.ssl-images-amazon.com/images/I/71uP4M30KnL.\_UX679\_.jpg" height="300"
18. alt="Moto Jacket" property="image">
19. <**p** property="description">100% Genuine Lambskin Leather - Kindly verify your size <**br**>with our size
20. chart to avoid any
21. return/exchange hassle. Order +1 size if you prefer loose fitting.</**p**>
22. </**header**>
23. <**hr**>
24. <**h2** class="text-info">Product details:</**h2**>
25. <**ul**>
26. <**div** property="offers" vocab="https://schema.org/" typeof="Offer">
27. <**li**>
28. <**strong**>Availability</**strong**>
29. <**meta** property="availability" content="https://schema.org/InStock">
30. In-Stock
31. </**li**>
32. <**li**><**strong**>Price:</**strong**> $
33. <**meta** property="priceCurrency" content="USD">
34. <**meta** property="price" content="150">
35. 150.00</**li**>
36. <**li**><**strong**>Used Condition:</**strong**>
37. <**meta** property="itemCondition" content="https://schema.org/NewCondition">
38. New</**li**>
39. <**li**><**strong**>Seller:</**strong**>
40. <**span** property="seller">Axme</**span**>
41. </**li**>
42. </**div**>
43. <**li**><**strong**>Material:</**strong**>
44. <**span** property="material">Leather</**span**>
45. </**li**>
46. <**li**><**strong**>Color:</**strong**>
47. <**span** property="color">Brown</**span**>
48. </**li**>
49. <**li**><**strong**>Mpn:</**strong**>
50. <**span** property="mpn">987689</**span**>
51. </**li**>
52. <**li**><**strong**>SKU:</**strong**>
53. <**span** property="sku">DM9201</**span**>
54. </**li**>
55. <**li**><**strong**>Brand:</**strong**>
56. <**span** property="brand">FUBU</**span**>
57. </**li**>
58. </**ul**>
59. <**section**>
60. <**h2** class="text-info">Sample reviews ★★★★★</**h2**>
61. <**h4** property="aggregateRating" vocab="https://schema.org/" typeof="AggregateRating">
62. Rating: <**span** property="ratingValue">4.5</**span**>
63. out of <**span** property="bestRating">5</**span**>
64. stars ( <**span** property="reviewCount">156</**span**> ratings)
65. </**h4**>
66. <**article** property="review" typeof="Review">
67. <**h4** property="author">Gaston</**h4**>
68. <**time** datetime="2020-06-06T13:31:04+00:00">
69. <**meta** property="datePublished" content="2020-06-06T13:31:04+00:0">
70. June 6<**sup**>th</**sup**>, 2020 at 1:31</**time**>
71. <**p**>Review for: <**span** property="itemReviewed">
72. Brown Leather Jacket Men
73. </**span**> </**p**>
74. <**p** property="reviewRating" typeof="Rating">Rating:
75. <**span** property="ratingValue">5</**span**>
76. </**p**>
78. <**p** property="reviewBody">Excellent Jacket - light and well made! Has a strong scent of leather, it is light weight,
79. and well made.</**p**>
80. </**article**>
81. <**article** property="review" typeof="Review">
82. <**h4** property="author">Wayne Laursen</**h4**>
83. <**time** datetime="2020-02-09T13:31:04-08:00">
84. <**meta** property="datePublished" content="2020-06-06T13:31:04+00:0">
85. Feb 9<**sup**>th</**sup**>, 2020 at 1:31</**time**>
86. <**p**>Review for: <**span** property="itemReviewed">
87. Brown Leather Jacket Men
88. </**span**> </**p**>
89. <**p** property="reviewRating" typeof="Rating">Rating:
90. <**span** property="ratingValue">5</**span**>
91. </**p**>
92. <**p** property="reviewBody">The leather is soft and attractive. The zipper is reversed, but easy to get used to. The fit
93. is tailored, just the way I like it.</**p**>
94. </**article**>
95. <**article** property="review" typeof="Review">
96. <**h4** property="author">Oliver W</**h4**>
97. <**time** datetime="2020-07-16T13:31:04-08:00">
98. <**meta** property="datePublished" content="2020-06-06T13:31:04+00:0">
99. July 16<**sup**>th</**sup**>, 2020 at 1:31</**time**>
100. <**p**>Review for: <**span** property="itemReviewed">
101. Brown Leather Jacket Men
102. </**span**> </**p**>
103. <**p** property="reviewRating" typeof="Rating">Rating:
104. <**span** property="ratingValue">5</**span**>
105. </**p**>
106. <**p** property="reviewBody">I bought this vest for my sons birthday..The quality of the leather was wonderful soft and
107. well made</**p**>
108. </**article**>
109. </**section**>
110. </**article**>
111. </**main**>
112. </**body**>
113. </**html**>

**Output**



**Explanation about Code**

1. here we are using Product Schema With RDFa Lite for Web and
2. here vocab lo we need to specifiy the url, what scheme we are following ani
3. typeof lo we need to specify the what type of caterogry we are using ani for eg: Recipe a , FAQ a, Products a etc...
4. property lo we need to specify what the property we are using
5. these follow same rule as other JSON and micro data i.e if the Property is of a Specific form then open new vocab and typeof as we did in offer property of the Products
6. NOTE: we need to write all the property which we are using in our website then we need to test the code in the rich testing tool, if anything is missing the it will give warming or error, then we need to fill or add all the details in our scheme Object, there is no strict rule that these are imp, every time, we need to follow the Google guidelines(their official website and we need to prepare the scheme Object
7. For example in our Product example, the main mandatory property are name and Either review or aggregateRating or offers as per the Google guidelines, u may think know do I know that these 2 fields(name and images) are mandatory , plz check the google docs(<https://developers.google.com/search/docs/advanced/structured-data/product> ) they mentioned all the mandatory properties and recommened properties in their google docs  
   same rule is applicable for the all the hotel and products and organizations

**3) scheme Data using JSON-LD**

<!**DOCTYPE** html>

<**html** lang="en">

  <**head**>

    <**meta** charset="UTF-8" />

    <**meta** name="viewport" content="width=device-width, initial-scale=1.0" />

    <**meta** http-equiv="X-UA-Compatible" content="ie=edge" />

    <**link**

      rel="stylesheet"

      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

    />

    <**link** rel="stylesheet" href="styles.css" />

    <**title**>Event</**title**>

  </**head**>

  <**body** class="container mt-4">

    <**article**>

      <**header**>

        <**h1** class="text-info">The Adventures of Kira and Morrison</**h1**>

        <**p**>

          The Adventures of Kira and Morrison is coming to Snickertown in a

          can’t miss performance.

        </**p**>

        <**img**

          src="http://www.lacm.edu/wp-content/uploads/2017/03/kira-morrison-lacm-recital-march-17-2017-6451-uai-1800x1199.jpg"

          alt="Kira and Morrison performance"

        />

      </**header**>

      <**ul** class="mt-4">

        <**li**><**strong**> Start Date:</**strong**> 12 July, 2020 at 1.30PM</**li**>

        <**li**><**strong**> End Date:</**strong**> 12 July, 2020 at 11.00PM</**li**>

        <**li**><**strong**> Where:</**strong**> Snickerpark Stadium</**li**>

        <**li**>

          <**strong**> Address:</**strong**> 100 West Snickerpark Dr,Snickertown, 19019

          PA, US

        </**li**>

        <**li**><**strong**> Registration Price:</**strong**> $230</**li**>

        <**li**><**strong**> Availability:</**strong**> In-Stock</**li**>

        <**li**><**strong**> Performer:</**strong**> Kira and Morrison</**li**>

      </**ul**>

    </**article**>

    <**script** type="application/ld+json">

      {

        "@context": "https://schema.org/",

        "@type": "Event",

        "name": "The Adventures of Kira and Morrison12",

        "description": "The Adventures of Kira and Morrison is coming to Snickertown in a can’t miss performance.",

        "image": "http://www.lacm.edu/wp-content/uploads/2017/03/kira-morrison-lacm-recital-march-17-2017-6451-uai-1800x1199.jpg",

        "startDate": "2022-03-03",

        "endDate": "2022-03-03",

        "location": {

          "@type": "Place",

          "name": "Snickerpark Stadium",

          "address": {

            "@type": "PostalAddress",

            "streetAddress": "100 West Snickerpark Dr",

            "addressLocality": "Snickertown",

            "postalCode": "19019",

            "addressRegion": "PA",

            "addressCountry": "USA"

          }

        },

        "offers": {

          "@type": "Offer",

          "price": "230",

          "priceCurrency": "USD",

          "availability": "https://schema.org/InStock",

          "url": "http://google.com",

          "validFrom": "2020-06-12"

        },

        "performer": {

          "@type": "PerformingGroup",

          "name": "Kira and Morrison"

        },

        "eventAttendanceMode": "https://schema.org/OnlineEventAttendanceMode",

        "organizer": {

          "@type": "Organization",

          "name": "dev2proddev2proddev2proddev2proddev2proddev2proddev2proddev2proddev2proddev2proddev2proddev2prod",

          "telephone": "( 33 1) 42 68 53 00",

          "url":"www.google.com",

          "image": "http://www.lacm.edu/wp-content/uploads/2017/03/kira-morrison-lacm-recital-march-17-2017-6451-uai-1800x1199.jpg",

          "address": {

            "@type": "PostalAddress",

            "addressLocality": "Paris, France",

            "postalCode": "F-75002",

            "streetAddress": "38 avenue de l'Opera"

          }

        },

        "eventStatus": "https://schema.org/EventScheduled"

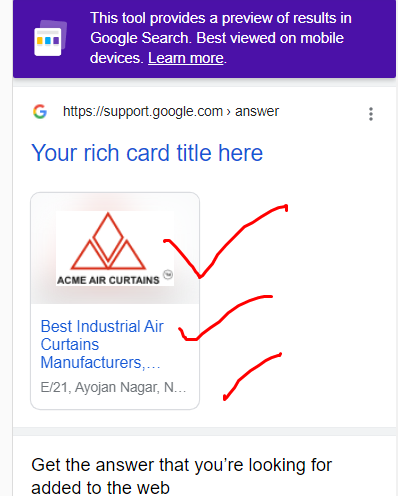
      }

    </**script**>

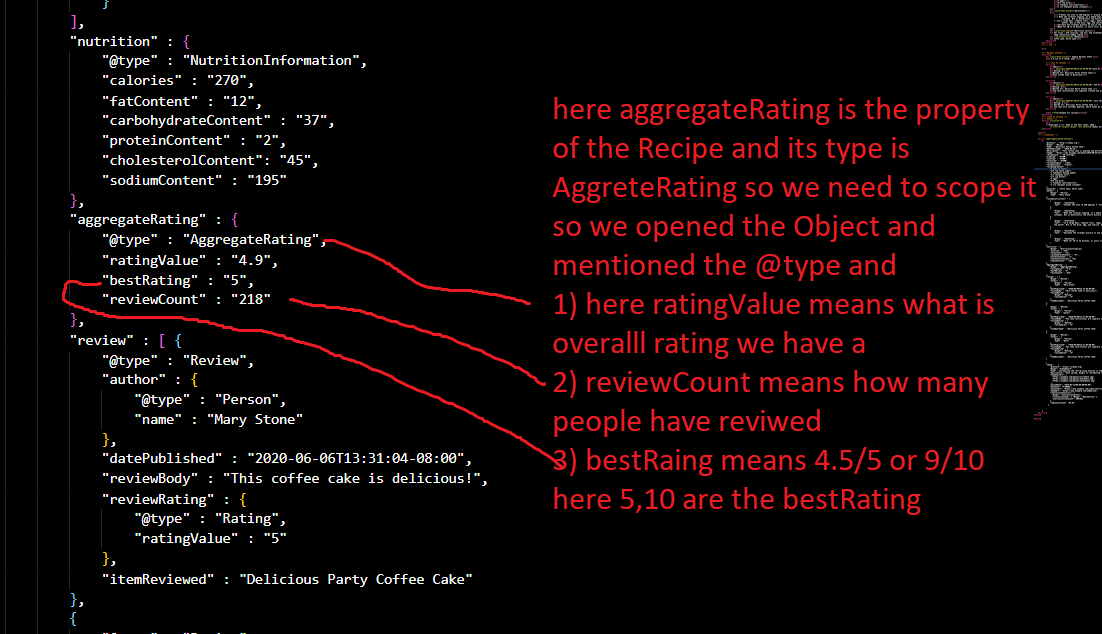
  </**body**>

</**html**>a

**Output**



**Explanation of the code**

1. @content specifies what object ur using ani example scheme.org(NOTE: there are so many other website which provide scheme object preparation so we need to specify the Object)
2. @type specifics which type thing ur using in ur code ani, like Recipe, FAQ, Events etc..
3. remaing alll the property we must need according to our website content
4. NOTE: there are some mandatory propery we need to write at anycost, while compiling these Scheme Object, google rich testing tool will suggest us and we need to write those alll properties
5. for example when we write location and came to know its a sperate entry, then we need to open the Object({}i.e flow brakets) and need to specify all the property which are mandatory for us in that file THEME when u encounter a sperate entry (like address,offer) we need to open the {} and write its property becoz their themselves are separate entities
6. 
7. These above explanation is not their in code( in json above real code example) I have showen u just to get idea that we use and give the rating in these way ani
8. For example in our Event example, the main mandatory property are location , location.address , location.url, name, startDate as per the Google guidelines, u may think know do I know that these 2 fields(name and images) are mandatory , plz check the google docs(<https://developers.google.com/search/docs/advanced/structured-data/event> ) they mentioned all the mandatory properties and recommened properties in their google docs  
   same rule is applicable for the all the hotel and products and organizations

NOTE:

I have not wrote all other scheme object type here, becoz their code is huge and no of pages of these doc will increase so im providing u a github link of mine, where I kept all the other scheme Object type there like FAQ,etc…  
<https://github.com/harishkumarforeverr/Scheme-Object-ps-Udemy>