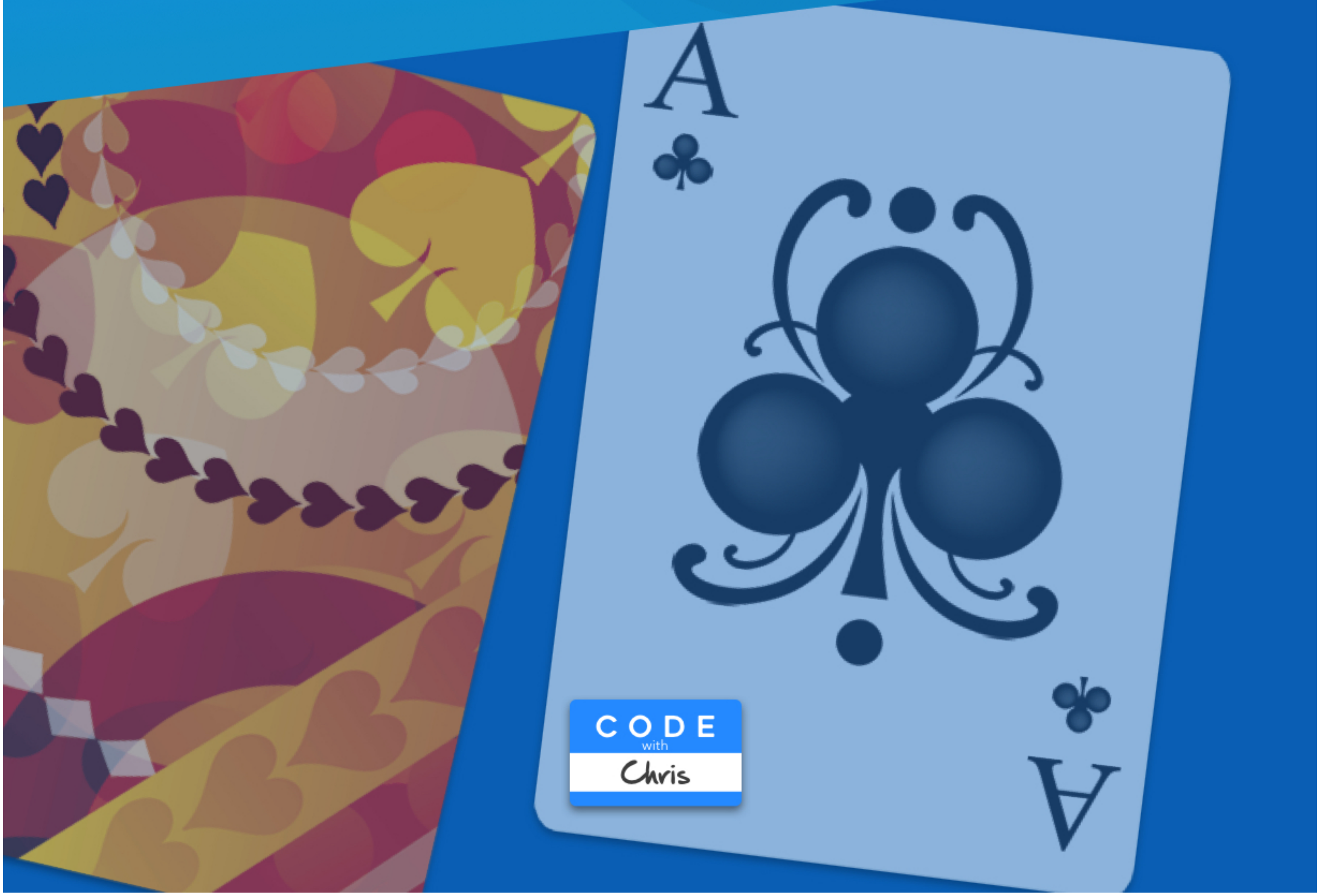
HOW TO MAKE AN APP FOR BEGINNERS

Lesson 4
First User Interface

RECAP NOTES



Key Concepts

1. As you're laying out your user interface, you'll come across two terms: **Safe Area** and **Margins**.

2. **Safe area** is the space that is guaranteed to be unobstructed by things like the "notch" or the rounded corners on the iPhone X series. If you place your elements within the "safe area", you can be sure they won't be covered by things such as the battery indicator, time etc.

3. By default when you specify positioning constraints for your elements, it'll be within the Safe Area. You have to manually open the dropdown and select "View" instead of Safe Area if you want your element to fall outside of the Safe Area (you'd want to do this for a full screen background, for example).

4. The **margins** are something that you can customize for your app and they just help you keep your elements within certain distance from the edges. There's a default amount of left/right margin.

5. When you're adding a new Auto Layout constraint, you'll see a checkbox to "**Constrain to Margins**". If this is enabled, you constraint will respect the margins.

6. The **UIImageView** element has a property called "**Content Mode**" which lets you change how the image is displayed inside.

7. The default is "**Scale to Fill**" which will stretch the image so that it fills up the entire image view.

8. Some other common options are "**Aspect Fit**" and "**Aspect Fill**".

9. **Aspect Fit** will increase the image size (maintaining the aspect ratio) to the maximum size that will fit inside the image view.

10. **Aspect Fill** will increase the image size (maintaining the aspect ratio) until the entire image view sill is filled up. This may result in some clipping of the image if the aspect ratio of the image and image view are different.

11. All the various screen sizes of iOS devices fit into one of several categories called **Size Classes**.

12. Size Classes are defined by a height and a width. The **width** and **height** can be either: **Regular**, **Compact** or **Any**.

13. For example the Size Class "Any Width, Any Height" (denoted by "**wAny, hAny**") includes all screen sizes and all orientations.

14. Another example, the Size Class "Compact Width, Compact Height" (**wC, hC**) is the smallest size class.

15. Size Classes exist so that you can fine tune your user interface for various screen sizes and orientations.

16. By default when you add elements onto your view and when you add auto layout constraints, you're adding them to the "Any Width, Any Height" (**wAny, hAny**) size class.

17. You can specifically add/remove elements and constraints for certain Size Classes. These are called **Variations**.

18. To see how this is done, check out the lesson video.