LESSON 21

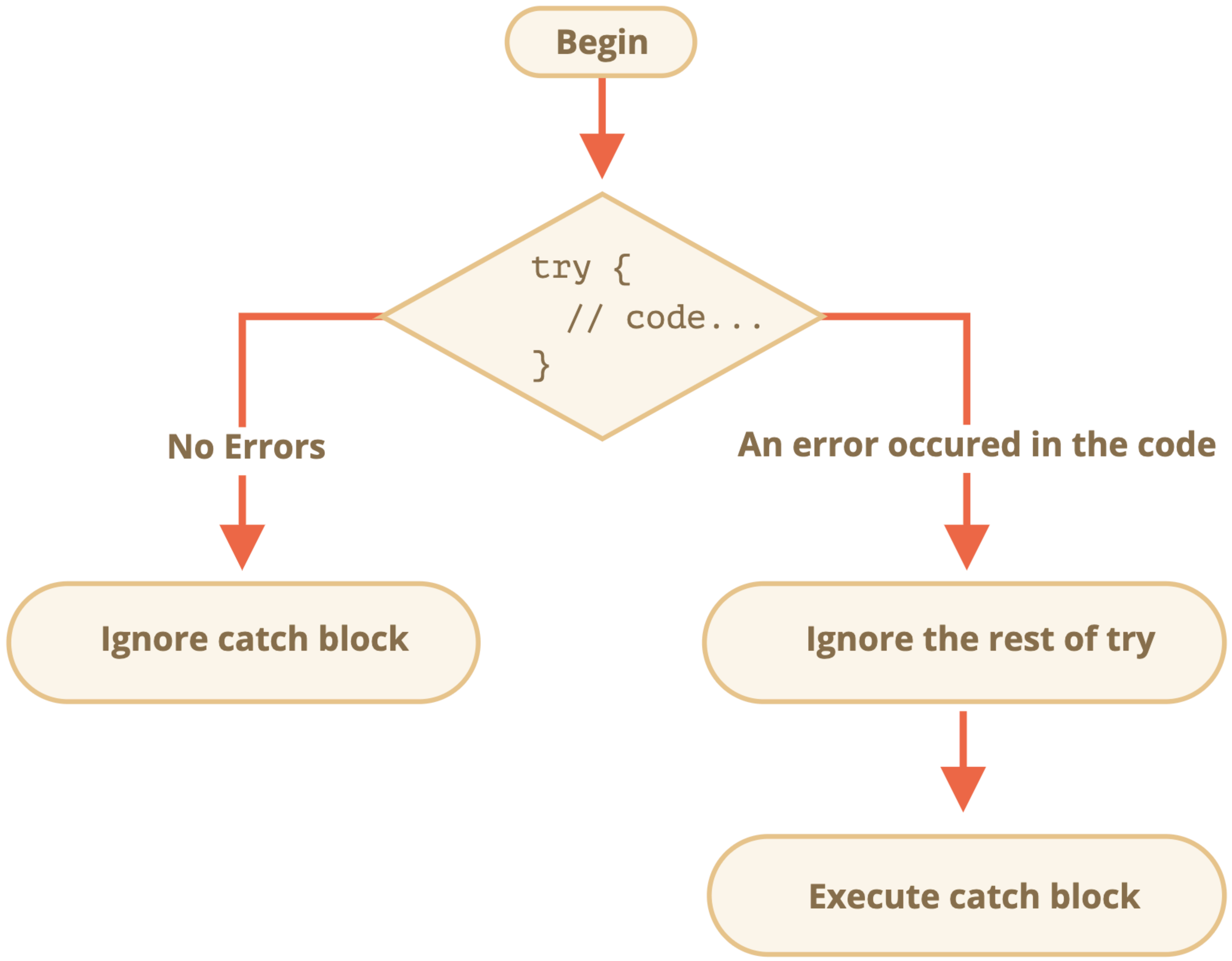
TRY…CATCH
ASYNC … AWAIT

ERROR HANDLING, "TRY..CATCH"

```
try {
  // code...
} catch (error) {
  // error handling
}
```

- only works for runtime errors
- `try..catch` works synchronously
- Get error is not necessary

HOW IT WORKS


Begin
try {
  // code...
}
No Errors
An error occured in the code
Ignore catch block
Ignore the rest of try
Execute catch block

ERROR OBJECT

- name
- message
- stack

- throw “error”
- new Error(“error”)

FINALLY

```
try {
  ...
} catch(e) {
  ...
} finally {
  ...
}
```

- try … finally

GLOBAL CATCH

```
window.onerror = function (message, url, line, col, error) {
  console.log(message);
};
```

ASYNC

- Returns promise
- Promise resolves that “return” passes

```
async function fn() {
  return "result";
}

async function() {
  return "result";
}

async () => {
  return "result";
}
```

AWAIT

- Synchronously waits until promise resolved
- Works only inside async function

EXAMPLE

```
async function fn() {
  return 42;
}

async function run() {
  console.log("start");
  console.log(await fn());
  console.log("finish");
}

run();
```

TEST

```
setTimeout(function () {
  console.log(1);
}, 0);

(async function fn() {
  return 2;
})().then((value) => {
  console.log(value);
});

const promise = Promise.resolve();

promise.then(() => {
  console.log(3);
});

console.log(4);
```

ASYNC + TRY/CATCH

```
async function fn() {
  if (confirm("resolve?")) {
    return "resolved";
  } else {
    throw new Error("rejected");
  }
}

async function run() {
  try {
    const result = await fn();

    console.log(result);
  } catch (error) {
    console.log(error);
  }
}

run();
```

ASYNC/AWAIT

```javascript
function timeout(message, time = 0) {
  return new Promise((done) => {
    setTimeout(() => done(message), time * 1000);
  });
}
function random(min, max) {
  return Math.round(Math.random() * (max - min + 1)) + min;
}
async function randomMessage() {
  const message = ["Hi!", "How are you?", "Are you ok?"][random(0, 2)];
  return timeout(message, 3);
}
async function chat() {
  const message = await randomMessage();
  console.log(message);
}
console.log("--- start ---");
chat().then(() => console.log("--- end ---"));
```