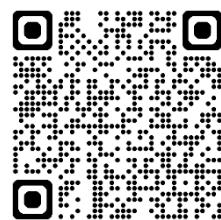


ChatGPT を使用して、先生の ChatGPT の書きたいことを HTML にさせてみた。

[2025/09/01, ChatGPT に作ってもらったときのチャット](#)



[2025/09/01, ChatGPT に作ってもらった HTML](#)



発表時の台詞 Ver1

台本(約 2 分 30 秒／ゆっくりめ)

【0:00～0:15|つかみ】

みんな、点数ってしんどいときあるやん？

上がったら嬉しいけど、下がった瞬間に「もうええわ」って手が止まる。今日は、その“止まり”を外す話をするで。

【0:15～0:45|問題提起】

評価(テストや順位)は必要やけど、十分ちゃう。

だって評価は「できた／できへん」を濃く見せる道具やから、失敗を避ける最適化が走りやすい。結果、挑戦が減る。これ、もったいないやん？

【0:45～1:30|主張と理由】

そこで“承認”。ここで言う承認は甘やかしやなくて、

事実ベースで「やったこと」を認める(例:ノートまとめた／質問できた)

人としての価値は点数で上下せえへん、を前提にする

この 2 つがあると、失敗が“損”やなく“燃料”に変わる。だから手がまた伸びる。

【1:30～2:05|具体策:明日からできる 3 つ】

① 前進ログ:授業の最後 1 分、「今日できた一歩」を一行で書く。

② ピア・リーディング:ペアで前進ログを読み合い、次の一步だけ返す。「ここ、明日これやってみよ」って一言で OK。

③ 再挑戦枠:ミスは回収可能って最初に決めとく。やり直せる設計にして、提出のハードル下げる。

【2:05～2:25|小話(ストーリー)】

この前「質問できた」って一言だけ書いた子がいたんよ。そっから連鎖して「エラー再現できた」「関数分けた」って増えて、教室がちょっとだけ賑やかになった。点じゃなくて“線”が見えた瞬間やった。

【2:25～2:45|結論・リクエスト】

評価はメーター、承認はガソリン。メーターだけ見てても進まれへん。

今日から一行でええから、前進ログつけてみよ。明日の自分の着手が、ちょっと速くなるから。

【2:45～3:00|しめ】

「できた／できへん」の間ににある“やった”を集めよ。

それが、みんなの学びを長持ちさせる一番の近道やで。ありがとう。

## 発表時の台詞 Ver2

### 導入(0:00~0:30)

「みなさん、点数や評価って、嬉しい時もあれば苦しい時もありますよね。今日お話ししたいのは、評価だけに頼らない『承認』の話です。」

### 問題提起(0:30~1:00)

「学校や職場では、点数やランキングで自分の価値を測る場面が多いです。でもそれだけだと、失敗を恐れて挑戦できなくなってしまいます。私自身も、『もうダメだ』と諦めた経験があります。」

### 解決の提案(1:00~2:00)

「そこで大切にしたいのが『承認』です。ここでの承認は、ただ褒めることではなく、やったことや存在そのものを認めることがあります。」

「具体的には、日々の小さな前進をメモしてみる、仲間と共有して励まし合う、失敗してもやり直す機会をつくる、などの方法があります。」

「このようにすると、点数だけでは見えない努力や成長が見えてきて、挑戦する勇気が湧いてきます。」

### まとめ(2:00~2:30)

「評価は方向を示してくれるメーターですが、承認は進み続けるための燃料です。今日からは、自分や周りの人の小さな一歩に目を向けてみませんか？」

### 締めくくり(2:30~3:00)

「最後まで聞いてくれてありがとうございます。みんなの学びや挑戦が、点数に左右されない豊かなものになりますように。」

## 発表時の台詞 Ver3

### 【0:00~0:20|導入】

「みなさん、今日は自分が作った Web ページの裏側を紹介します。このページは『評価から承認へ』というテーマで作っていて、1つの HTML ファイルの中に CSS と JavaScript も詰め込んでいます。」

### 【0:20~0:50|HTML の構成】

「まず HTML。<header> や <nav>、<main>、<footer> といったセクショニング要素を使って、見出しや目次、本文、フッターをきれいに分けています。記事部分は <article> と <section> を使って 4 種類の構成（序論・NVC・問題解決・ストーリー）を切り替えられるようにしています。キーボード操作やアクセシビリティも意識して、見出しに id を付けて目次からジャンプできるようにしてます。」

### 【0:50~1:30|CSS の工夫】

「次に CSS。--accent などのカスタムプロパティ（変数）で色や余白をまとめて定義して、ダークモードとライトモードの切り替えに対応しています。スマホでも読みやすいように、clamp() や @media クエリでフォントサイズやレイアウトをレスポンシブに調整。背景には color-mix() や radial-gradient() を使ってやわらかいグラデーションをかけて、見やすさと楽しさを両立しました。進捗バーや検索ハイライトも CSS で見た目を整えています。」

### 【1:30~2:10|JavaScript の役割】

「JavaScript では、テーマ（ダーク／ライト）やアクセントカラー、構成の切り替えをローカルストレージに保存して次回も同じ設定で開けるようにしています。また、IntersectionObserver を使ってスクロールに合わせて目次をハイライトしたり、入力したキーワードをリアルタイムで本文内からハイライトして、Enter キーで次の一致にジャンプできるようにしているのもポイントです。キーボードショートカットで構成を 1~4 キーで切り替えられるようにもして、操作性を上げました。」

### 【2:10~2:30|まとめ】

「まとめると、このページは 1 ファイルで HTML・CSS・JS を完結させつつ、テーマ切替や検索、目次連動など、ちょっとしたアプリ並みの機能を実装しています。普段の学習でもこうした仕組みを取り入れると、閲覧者の体験がぐっと良くなるので、ぜひ試してみてください。ありがとうございました！」